

COMPUTER AIDED
PROCESS CONTROL SYSTEMS SYNTHESIS
USING RULE-BASED PROGRAMMING

A Thesis
submitted in fulfilment of the requirements for the Degree
of
Doctor of Philosophy in Chemical Engineering
in the
University of Canterbury
by
C. J. Williamson

University of Canterbury

1989

Acknowledgements

I would like to thank my supervisor, Dr Brian Earl, for supplying the generous support that has enabled me to complete this project. Thanks are also due to Dr Raj Gupta who assisted me early in the project to unravel the complexities of structural controllability which subsequently played an important part in this work. I am grateful to the University of Canterbury for providing me with welcome financial assistance in the form of a Canterbury Postgraduate Scholarship.

Thanks to my Postgraduate colleagues whose friendly company and occasional advice have made my stay in the Department a pleasant one. Finally, and most importantly, I thank my parents for their generous encouragement and patience over the past few years.

Table of Contents

Abstract

Chapter 1 - Introduction.....	I-1
1.0 Introduction.....	I-1
2.0 Thesis Structure	I-2
3.0 Previous Research into Control Systems Synthesis	I-4
4.0 Previous Research into Control-in-Design Techniques.....	I-9
5.0 Control Systems Synthesis Using Expert Systems	I-13
6.0 Conclusion - Piping and Instrumentation Diagram Generation	I-23
7.0 Nomenclature.....	I-23
Chapter 2 - Distillation Control.....	II-1
1.0 Introduction.....	II-1
2.0 Distillation Control.....	II-1
3.0 Expert System For Distillation Column Control Systems Synthesis.....	II-20
4.0 Nomenclature.....	II-28
Chapter 3 - Heat Exchanger Control	III-1
1.0 Knowledge Acquisition For A Heat Exchanger Expert System.....	III-1
2.0 Heat Exchangers Without an Entire Stream Changing Phase	III-1

3.0	Heat Exchangers With A Phase Change	III-6
4.0	Expert System for Heat Exchanger Control System Synthesis.....	III-12
5.0	Nomenclature.....	III-17
Chapter 4 - Reactor Control.....		IV-1
1.0	Knowledge Acquisition for a Reactor Control Expert System	IV-1
2.0	Continuous Stirred Tank Reactor Control	IV-3
3.0	Plug Flow Reactors.....	IV-8
4.0	Fixed Bed Reactors.....	IV-11
5.0	Industrial Case Study of The ICI Oxo Reactor.....	IV-12
6.0	The Reactor Control System Synthesis Expert System	IV-14
7.0	The Program at Work	IV-21
8.0	Nomenclature.....	IV-23
Chapter 5 - Structural Controllability		V-1
1.0	Introduction.....	V-1
2.0	State Controllability.....	V-1
3.0	Structural Controllability.....	V-3
4.0	Structural Controllability Applied to Control Systems Synthesis.....	V-5
5.0	Functional Controllability.....	V-11
6.0	Conclusion.....	V-13
7.0	Nomenclature.....	V-14

Chapter 6 - Whole Plant Control System Synthesis	VI-1
1.0 Whole Plant Control System Synthesis Using Expert Systems	VI-1
2.0 Preliminary Control System Synthesis Strategies.....	VI-2
3.0 The Control Systems Synthesis Package.....	VI-3
4.0 A Control System For a Complete Process	VI-6
5.0 Conclusion.....	VI-20
6.0 Nomenclature.....	VI-21
Chapter 7 - Conclusions and Recommendations For Future Work	VII-1
1.0 Structural Controllability	VII-1
2.0 Individual Unit Operation Expert Systems.....	VII-2
3.0 The Whole Plant Control Systems Synthesis Program.....	VII-5
4.0 Impact of Problems and Improvements in Expert SystemsTechnology	VII-6
References.....	VIII-1
Appendix AI - Interaction With The Control System Synthesis Package.....	AI-1
Appendix AII- Duff 's Algorithm and the Johnston and Barton Alternative.....	AII-1
Appendix AIII - Rule Base For the Distillation Expert System.....	AIII-1
Appendix AIV - Rule Base For The Heat Exchanger Expert System.....	AIV-1
Appendix AV - Rule Base For The Reactor Expert System	AV-1
Appendix AVI - TurboProlog Listing For The Distillation Expert System.....	AVI-1

Appendix AVII -	TurboProlog Listing For The Heat Exchanger Expert System.....	AVII-1
Appendix AVIII -	TurboProlog Listing For The Reactor Expert System.....	AVIII-1
Appendix AIX -	TurboProlog Listings of the Extra Programs For The Coordinated Expert Systems.....	AIX-1

Abstract

Current steady-state process simulators have greatly increased the speed and efficiency of the development of Process Flow Diagrams. Chemical Engineers would benefit in the same way from a Computer Aided Design package to assist with generating completed Piping and Instrument Diagrams.

Despite the many theoretical methods available in the control science area there is no single and complete available solution to the problem of synthesising control systems for whole chemical processes and therefore no concrete basis from which to develop a computer program. Design activities rely on a significant experience factor and this element has largely been ignored especially in control systems synthesis. The recent emergence of rule-based programming allows this "experience" dimension to be added to software. Although there is previous work in the literature on expert systems for distillation column control systems synthesis there is very little published on programs for other unit operations or the whole plant problem.

In this project the problem of how to set up an expert system for whole plant control systems synthesis was addressed. As a preliminary step this required that expert systems for control systems synthesis for unit operations be written. The necessary knowledge to do this for distillation columns, heat exchangers and reactors was sourced from the literature and programs developed for each using a shell written in a version of Prolog. These programs were coordinated to work together and provide controllable solutions to whole process control problems using a matrix representation of the relationship between control objectives and manipulated variables developed in structural controllability analysis. This provided the framework for a prototype whole plant program. The operation of all the programs is illustrated using typical examples and their rulebases included in appendices to the thesis.

The work demonstrated that, with more extensive rulebases than it was possible to develop in the time available for this project including access to theoretical methods when required, expert systems could provide a useful solution to both unit operation and whole plant control systems synthesis problems.

Chapter 1 - Introduction

1.0 Introduction

Computer Aided Design has become progressively more important in the chemical engineering design office. The past few years have seen a proliferation of sequential modular steady-state process simulators and at least one equation-based package (SPEEDUP). As yet, however, there is no available software that an engineer can use to aid in the synthesis of control systems for whole chemical processes. There is a need for a package that allows the rapid synthesis of control systems for process alternatives thereby making the control aspect an integral part of the design process.

Rule-based programming, the major subset of expert systems technology, has grown to maturity in the last few years. This has allowed engineers to experiment with adding an "experience" dimension to their software. However, although this has led to expert systems finding commercial use in areas such as process malfunction diagnosis and process control their use as synthesis tools remains a research area.

Control systems synthesis is a many faceted problem encompassing aspects of both design and control. This complexity is the main reason why a successful solution has eluded researchers. There are combinatorial difficulties, which often accompany design problems, in the pairing of control objectives and manipulated variables. For example, a distillation column's five control objectives and five manipulated variables can be paired up in 360,360 different ways if ratios between two manipulated variables are allowed as alternatives (Shinskey, 1984). A rigorous optimisation within this group would be expensive and unnecessary. Control quality and stability for these possibilities are important criteria in choosing the best solution and must be considered. There are a number of theoretical tools that are useful in solving this part of the problem but at the same time none of them represents the complete picture. Industrial control experts use experience to find workable solutions to control problems without using complex theoretical tools. The usefulness of this experience factor is recognised in process synthesis but has largely been ignored in control systems synthesis. Research workers have instead approached the problem with analytical techniques and have spent far less time on the contribution made by

heuristics. It remains an unresolved question whether the solution to developing a package for control systems synthesis lies in an integration of presently available analytical methods and heuristics or in some as yet undiscovered theoretical technique, although the theoretical solution still seems a long way off.

As the solution to the problem seems closest using rulebased programming and already developed theoretical techniques this project aimed to explore the contribution to computer aided whole plant control system synthesis possible using rulebased programming. The research addressed a number of specific objectives within this area;

1) As heuristic rules have already proven useful in reducing the combinatorial difficulties associated with process design problems (Lien, 1987) to establish whether this was also true in control systems synthesis.

2) To identify relevant heuristic rules for control systems synthesis and investigate how best to translate them into current expert system software.

3) To investigate where heuristic methods should be used in preference to available theoretical methods and to research the integration of heuristic and theoretical methods into a single package for control systems synthesis.

4) To identify any parts of the control systems synthesis problem that can be handled effectively only by using heuristic methods.

5) As control systems synthesis knowledge is directed at a unit operations level to investigate how expert systems developed from this should be coordinated when an entire plant, rather than isolated units, is considered.

Knowledge on the control of three key unit operations; distillation columns, heat exchangers and reactors was acquired from the literature and organised into the appropriate form to write the corresponding expert systems. After they were completed in isolation these individual unit operation expert systems had to be coordinated together to solve whole plant control systems synthesis problems. Previous research into Structural Controllability Analysis, a technique previously considered for control systems synthesis, presented a possible solution to this part of the problem which allowed the development of a prototype package for whole processes.

2.0 Thesis Structure

The thesis is written in seven chapters. Each has a specific symbology which is explained in a nomenclature section at the end of the chapter.

Chapter 1 - The rest of this chapter is largely devoted to a literature review of previous research in the control systems synthesis and control-in-design areas. There is also an introduction to expert systems included to background terms and approaches used later in the thesis.

Chapter 2 - The first part of the chapter summarises the knowledge about distillation control found in the literature. It includes discussion of mass balance, single composition and dual composition control schemes. There are also details of further extensions to the basic regulatory structure of the column control system such as feed forward and cascade additions. The second part of the chapter describes the "shell" written in Prolog to implement this knowledge as an expert system and gives examples of the performance of the program.

Chapter 3 - The majority of the chapter describes the knowledge collected on heat exchanger control for two classes of exchanger. Heat exchangers without a phase change in a stream and those with a completely condensed heating stream. The second and smaller part of the chapter describes the expert system for heat exchanger control developed from this knowledge. A similar programming style to the earlier work on distillation was used and examples from the two exchanger classes are included to demonstrate program operation.

Chapter 4 - This chapter deals with the control of Continuous Stirred Tank, Tubular and Fixed bed reactors. The results of the literature survey and the consequent expert system are discussed. There is an example of the program at work on the problem of an exothermic CSTR.

Chapter 5 - A detailed review of Structural Controllability analysis makes up this chapter because of its importance in the development of the package for whole plant control system synthesis. Useful concepts and tools important in the final stage of the research, especially the work of Johnston and Barton, are highlighted and the fundamental necessity that a process control system satisfy structural controllability tests emphasised.

Chapter 6 - Describes the package for whole plant control systems synthesis using expert systems for unit operations coordinated by ideas taken from structural controllability. The solution provided by the program on an example process is included and compared with a control system for the same example derived from structural controllability considerations alone.

Chapter 7 - Conclusions drawn from the current work and some recommendations for the future make up this chapter.

There are a number of appendices including English translations of the rulebases used in all the expert systems, the listings of the Prolog programs, the complete interaction between user and program that occurred when solving the whole plant problem in chapter 6 and a comparison between different mathematical methods required to provide the necessary structural controllability information for control system's synthesis.

3.0 Previous Research into Control Systems Synthesis

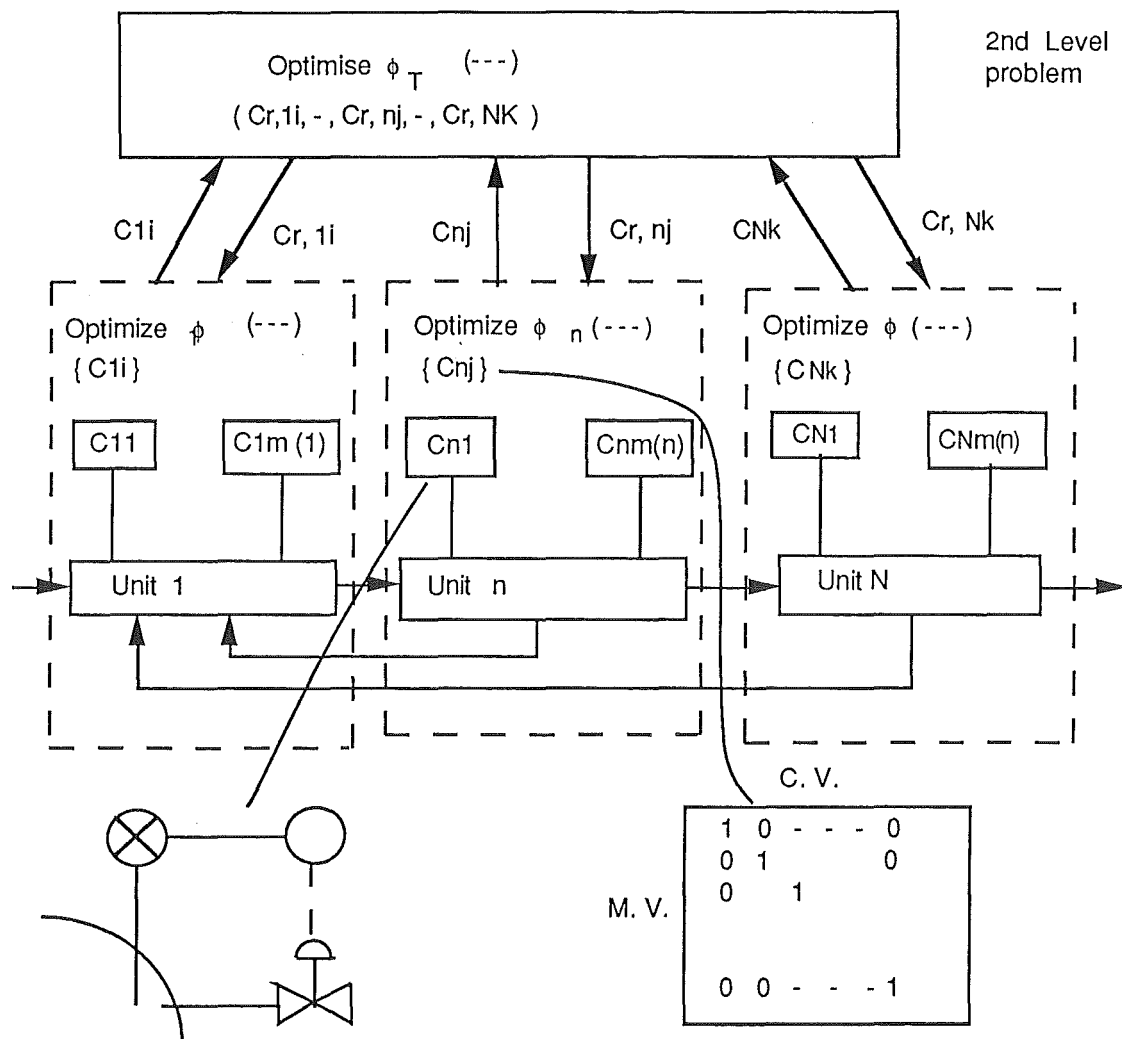
Control systems synthesis for whole chemical plants is a many faceted problem and the complete organisation of the solution method is as yet unresolved. The historical perspective is provided by Buckley (1964). He recommended material balance control to regulate plant production rate. The unit operations in the process are level controlled using either the product or feed streams. Any change in feed rate propagates through the plant. The unit operations quality control structure is superimposed on this to complete the control system. The assumption is that the feed changes are made infrequently and slowly while quality control disturbances occur fast and often. The material balance system doesn't therefore interact with the quality control system and both can be designed in isolation. This approach was adequate when plants were designed with little or no recycle, heat integration or other sources of interaction between unit operations. Now chemical plants are more integrated and energy efficient, significantly increasing the interaction between plant units. This change has emphasised the need to consider control early in the design process. The logical tools to accomplish this are computer programs that diagnose control problems in process designs and allow the rapid generation of whole plant control systems from process flow diagrams.

The first of the new generation of published work in this area was by Umeda et al (1978) who suggested a two level synthesis technique. The flowsheet is divided into unit operations and possible control systems for each are generated by analysis of the degrees of freedom available in mass, energy and momentum balances. These possibilities are screened using heuristic rules initially and finally through steady state and dynamic simulation to arrive at a collection of control systems that optimise the control performance of individual units and make up a first estimate of a control system for the flowsheet. This system is then analysed at the second level to eliminate conflicts between unit control schemes by applying heuristic rules with respect to the following;

- Difference in hold-up volume of related units
- Intermediate storage for isolating related units

-Location of flowrate control

In order to optimise the overall control performance of the process the revised control scheme is then passed back to the first level for reevaluation. The synthesis is complete when the proposed system passes through the procedure without being altered at the second level. Stephanopoulos has used a similar technique as an example in his text on process control (1984). A mathematical statement of the procedure is illustrated in Fig. 1.



C. V. : A set of control variables

M. V. : A set of manipulated variables

Fig. 1 Mathematical Statement of the Two Level Synthesis Method

The control performance functions at the unit level are labeled $\phi_1 \dots \phi_n$ and at the plant level, ϕ_T . Individual loops within the n th unit are $C_{n1} \dots C_{nm}$ and the collection of loops for the n th unit is designated C_{nj} . This is a matrix of controlled variables vs. manipulated variables with entries where a manipulation is connected to a controlled variable in a loop. The revised control systems at the second level are $C_{r,nj}$ and these are returned to the first level as shown.

Control systems synthesis was subdivided into five aspects by Nishida, Stephanopoulos and Westerberg (1981);

- a) A complete definition of control objectives for the process
- b) The selection of controlled variables
- c) The selection of a measured variable set
- d) The selection of a manipulated variable set
- e) The design of the control structure (the interconnections between the measured and manipulated variables)

There are a number of techniques to aid with choices required in these five criteria. The identification of controlled variables, for example, can be divided into four distinct classes;

- i) Operational constraints, usually in the interests of safety or process requirements eg. a pressure or temperature must be kept below allowable maximum values or within particular bounds to achieve a significant reaction yield, levels and flowrates must be controlled to adjust plant throughput.
- ii) Product quality requirements eg. a 99% pure product.
- iii) Environmental regulations that require that the level of contamination of waste streams be kept below some maximum value.
- iv) Economic considerations ie. which controlled variables should be used to decide the most profitable operating point for the plant.

It is generally straightforward to identify controlled variables that satisfy the first three classes of objectives but more difficult to identify those in the fourth category. Industrial practice is to solve the optimum steady state control problem on-line and to alter set-points or adjust manipulated variables to achieve this solution. Morari, Arkun and Stephanopoulos (1980) compared the minimised operating costs of a completely optimised process with the operating costs of the same process with key

variables kept constant by feedback. If the economic deterioration between the two is small the feedback alternative can be used. This simplifies on-line optimisation considerably.

Fisher, Doherty and Douglas (1988c) developed a similar but approximate approach to identify the optimisation variables as part of their steady state controllability analysis. If a variable hardly changes at the optimum operating conditions as steady state optimisation studies are made for a range of process disturbances then keeping it constant by feedback will approach the economic optimum. The optimisations are done by a short-cut rather than rigorous approach.

Govind and Powers (1982) pioneered an approach based on establishing all the possible measured and manipulated variables in two steps and combining these groups in a third to produce control structures for the process. The controlled variables must be identified before this procedure begins;

- 1) Possible methods for identifying the controlled variables are generated using a structural system array of the process. The array is a non-numerical representation of the mass, energy and momentum relationships between flowsheet variables in the Laplace domain. If a controlled variable appears in an equation in the array then all the variables apart from the controlled variable must be measured to allow calculation of it. The lower level variables then become the constraints and the process is repeated until no more branches can be added to the tree of measured variable set possibilities. This method is repeated for all controlled variables to identify the possible measured variable sets.

- 2) Manipulated variable sets are produced from the system cause and effect graph (a diagrammatic representation of the dynamic relationship between process variables). A directed edge points from one node to another if that variable affects the other. A variable is suitable to alter another if it affects it, and if it can be successfully screened through a set of heuristic rules and the constraint-variable transfer function (modeled as first-order plus deadtime) has an acceptable gain, time constant and lag.

- 3) The solution sets to the first two steps can be combined in different ways to produce a number of alternative control structures (feedback, feedforward and cascade) for any controlled variable and these can be grouped together to make up the control structure for the system. The method doesn't analyse further to establish which of the proposed structures is the best.

Morari and Stephanopoulos (1980) identified a basic weakness in this approach. The suggested structures could be uncontrollable. They developed a

procedure, structural controllability analysis, that identified all feasible manipulated variable sets without error. Their method was refined by other researchers and is explained more fully in chapter 5.

The manipulated variable sets can be compared using measures derived from the transfer function matrix relating inputs to outputs, G . Singular Value Analysis has attracted recent research interest. The singular values are the square roots of the eigenvalues of the matrix G^+G . A number of researchers have demonstrated that useful conclusions about control quality can be drawn using these quantities. Johnston and Barton (1985b), for instance, used the following to compare different manipulated variable sets for a double effect evaporator ;

- i) $\sigma(\min)[I + GK]^{-1}$ - this should be large for good quality control in the face of disturbances
- ii) $\sigma(\max)[GK(I+GK)^{-1}] \approx \sigma(\min)[GK(I+GK)^{-1}] \approx 1$ for good set-point tracking
- iii) $\sigma(\min)[G]$ large to prevent manipulated variable saturation
- iv) The process condition number, $\gamma = \sigma(\max)/\sigma(\min)$, should be small to give stability in the event of model/plant mismatch.

I The identity matrix

G The plant transfer function matrix

K The matrix of controller gains

Graphs of the four quantities (i - iv) vs. disturbance frequency are used to characterise and compare different control structures. This approach is suggested as an alternative to dynamic simulation.

Once the manipulated and measured variable sets are identified the interconnections between them must be found. The accepted approach in the last 20 years has been to minimise interaction between Single-Input Single-Output (SISO) loops and Bristol's relative gain array has proved a useful tool in this respect. More recently singular value decomposition (Lau, Alvarez and Jensen, 1985) has been investigated as a more powerful alternative.

The final stage in the design of a control system involves adding improvements to the control scheme such as constraint control, overrides, variable structure, cascade and feed-forward loops and alarm systems. These are added taking into account the disturbances, start-up and shutdown procedures and safety

requirements of the plant. There are few if any applicable theoretical techniques for selecting these additions. A comprehensive review of much of the work described in this section is provided by Stephanopoulos (1983).

4.0 Previous Research into Control-in-Design Techniques

Recent research has tried to find tools that include control considerations in the design process as well as improving control system synthesis. There are two separate paths with the same ultimate goal, to synthesize as easy-to-control a design as possible. The first uses short-cut methods and the second has attempted to formulate a more analytical framework to attack the control in design problem. A summary of these approaches, discussed in the next few paragraphs, is illustrated in Fig. 2.

Fisher, Doherty and Douglas's procedure (1988a, 1988b, 1988c) belongs to the first as it is reliant on short-cut calculation techniques to screen possible flowsheets at a preliminary design stage. Quoting from the first paper in the series (1988a);

" At the preliminary stage of design, the optimum steady-state designs of various process alternatives are often uncontrollable ie. there are not enough manipulative variables in order to satisfy the process constraints and to optimise all the operating variables. Controllability can be restored by (1) modifying the flow sheet to add more manipulative variables, (2) overdesigning certain pieces of equipment so that the process constraints never become active for the complete range of process disturbances, or (3) ignoring the optimisation of the least important operating variables. The goal of a controllability analysis is to determine which of these alternatives has the smallest cost penalty"

In their second paper in the series they consider process operability. Quoting once again from their work (1988b);

" As disturbances enter the plant the fixed equipment sizes may prevent the process constraints from being satisfied or may prevent the operating variables from being adjusted to significantly lower the operating costs. Operability problems can be overcome either by an appropriate amount of flexibility or by developing alternative operating policies, and we want to determine the alternative that has the smallest cost penalty "

The third stage in their method concentrates on identifying key optimisation variables for control. Quoting from the third paper in the series (1988c);

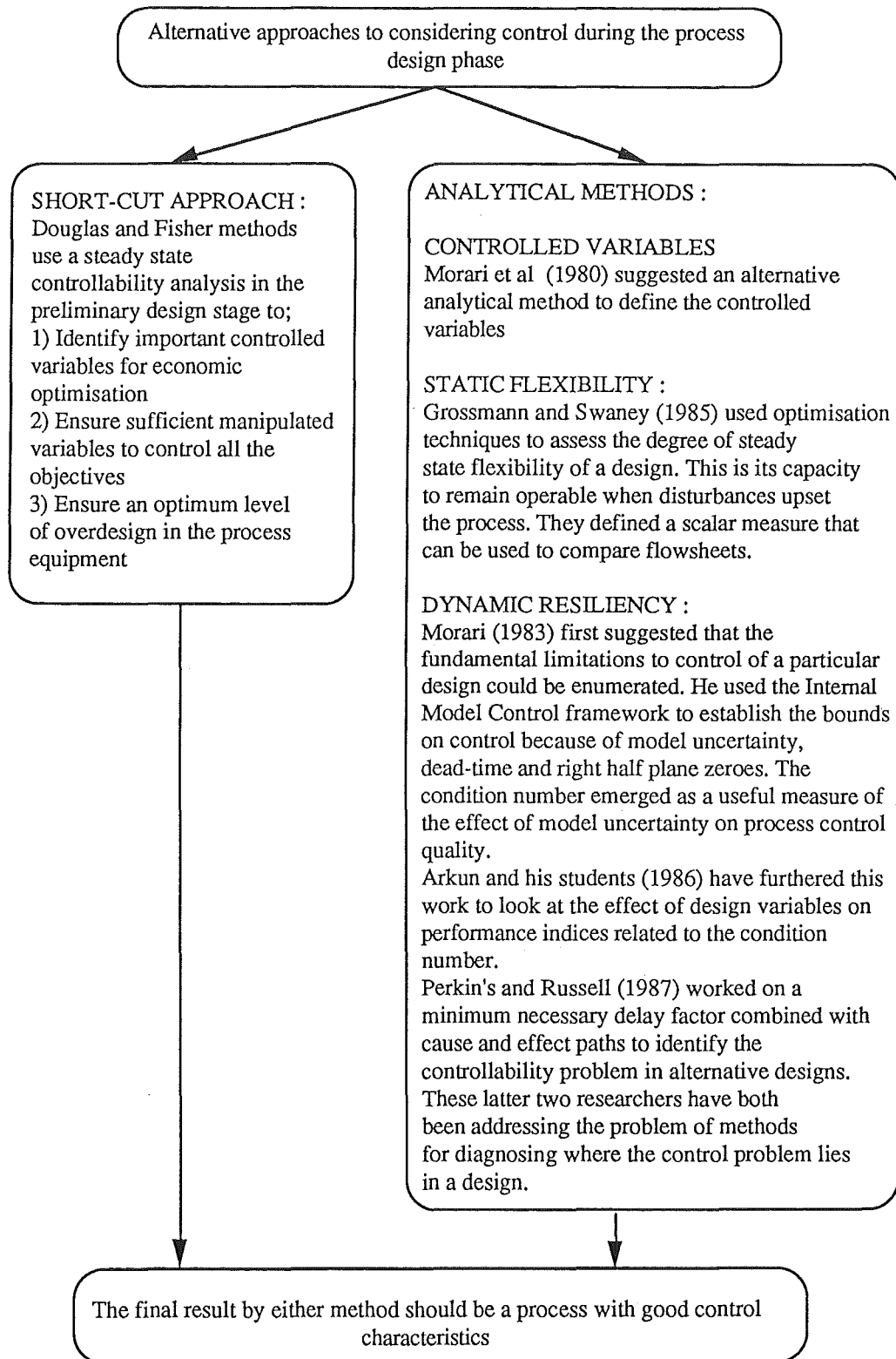


Fig. 2 Control In Design Methods

" By solving the optimum steady-state control problem in terms of the significant disturbances and manipulative variables, we often find that the optimum values of *some* of the operating and/or manipulative variables lie at constraints. If we select these constrained variables as controlled variables, the resulting feedback system will have near optimal performance without the need for measuring all the disturbances or for calculating the entire optimum steady-state control policy on-line"

Their whole synthesis procedure is summarised in Table 1. The discussed methods make up level 1 of the complete approach.

level 1: steady-state considerations

- 1a. Controllability. Identify the economically significant disturbances, and ensure that there are an adequate number of manipulative variables in order to be able to satisfy the process constraints and to optimize the operating variables over the complete range of the anticipated disturbances.
- 1b. Operability. Ensure that there is close to the optimum amount of overdesign to be able to satisfy the process constraints and to minimize the "expected" operating costs for the complete range of anticipated disturbances.
- 1c. Select the controlled variables. Select a set of controlled variables so that the steady-state operating costs will be essentially minimized.
- 1d. Steady-state screening of control structures. Assess the amount of interaction in alternative control structures.

level 2: normal dynamic operation—small perturbations from steady state

- 2a. Inventory control. Ensure that the plant material and energy balances can be closed, and assess the need for intermediate storage capacity.
- 2b. Dynamic control. Assess the stability of the control structure alternatives, and ensure robustness. The analysis includes flow-sheet modifications (e.g., additional overdesign) to ensure process operability in the dynamic state.

level 3: abnormal dynamic operation

- 3a. Start-up and shut-down. Assess the need for special control systems for the start-up and shut-down of the plant.
- 3b. Diagnostics and failure recovery. Ensure safe operation when equipment failures are encountered.

level 4: implementation

- 4a. Distributed control. Organize the levels of local unit control, plant control, and supervisory control.
- 4b. Human interface. Ensure that the operators can operate the plant.

Table 1. Hierarchical Approach to Control Systems Synthesis
(After Fisher, Doherty and Douglas)

Swaney and Grossmann (1985) investigated a flexibility index which can be used to compare different processes. This scalar index, is a measure of the size of steady-state disturbance a design can withstand and still be operable. Unit operations are part of the control structure. If they saturate in the face of disturbances control is impossible. The approach is an analytical alternative to the Fisher and Douglas work on process operability but is far more computationally intensive.

There have been a number of research efforts aimed at measures of dynamic resiliency. These ideas first came from Morari (1983) who tried to develop a theoretical framework to find the bounds on control possible for a design regardless of the controller type. He used the Internal Model Control (IMC) framework to come up with various measures indicative of particular control properties. He suggested three fundamental limitations that prevent implementing the plant transfer function inverse as the controller and realising perfect control (1) Non-minimum phase elements ie. deadtime and right half plane transmission zeros (these cause inverse response) (2) Physical constraints on manipulated variables (3) Model/Plant mismatch. In order to understand the control behaviour of non-minimum phase plants he factorised the transfer function matrix into two parts: $G_-G_+ = G$. The G_- part is invertible and represents the best IMC controller while G_+ becomes the closed loop transfer function. The optimum factorisation minimises an error measure (Integral Square Error) and represents the best control possible. The optimum factorisation for a process with time delays is a matrix with exponential terms on the diagonal representing the minimum achievable delay in output response. As an example, for a system with the optimum factorisation;

$$G_+ = \begin{pmatrix} e^{-s} & \# \\ \# & e^{-3s} \end{pmatrix}$$

there is no control system that would achieve a setpoint with less than a single time unit delay in output 1 and a 3 time unit delay in output 2. The magnitude of these fundamental time delays are therefore a measure for comparing alternative flowsheets or a means to suggest design changes that improve their value. An example of a comparison between different control methods for heat exchanger networks using this approach can be found in the literature (Holt and Morari, 1985).

The minimum singular value is a measure of the tendency of the plant manipulated variables to saturate. Morari (1983) proved that

$$\|y_s - d\| = \sigma_m(G) \|u\|_{\max} \quad (1)$$

$\|y_s - d\|$ The norm of the vector of disturbed outputs

$\sigma_m(G)$ The minimum singular value of the transfer function matrix

$\|u\|_{\max}$ The norm of the vector of maximum input values

From (1) it is clear that the larger the minimum singular value of a process transfer function matrix for a multivariable system the less likely the manipulated variables are to saturate. This matrix quantity therefore represents a bound on the disturbances that a process can handle before an input reaches its maximum value.

The condition number emerged as a useful measure of the sensitivity of designs to mismatches between model and plant. The condition number, the ratio between the maximum and minimum singular values, measures the closeness to singularity of a matrix. As the condition number increases from 1 the matrix can be regarded as progressively less well conditioned until, if it is infinite, the matrix is singular.

Recent research on the "robustness" of control systems (Morari, 1983) has shown that the condition number is a quantitative measure of the sensitivity of the process to variations between the model used for controller design and the actual plant. More ill-conditioned processes rapidly lose control performance when operating conditions move away from the design point if an inexact model is used and may even become unstable. Barton et al (1986) describe the use of the condition number to screen ore recovery flowsheets. The results of the study, supported by dynamic simulation, showed that the condition number provided a convenient and accurate quantitative measure for the comparison of candidate flowsheets on the basis of control difficulty. Other studies of this type have been made with similar success (Perkins and Wong, (1985) and Levien and Morari (1987)).

The condition number concept has been extended by Arkun (1986) to define controllability performance indices that provide some diagnosis capability into which variables need to be changed to improve controllability. Russell and Perkins (1987) also recognised the failure of the developing techniques to diagnose exactly which elements of a design are causing controllability problems. They studied the "minimum necessary delay", a concept for grading systems with time delays, and combined it with cause and effect pathways in system matrices to identify the state variable and output responsible for it.

5.0 Control Systems Synthesis Using Expert Systems

A precisely stated synthesis procedure doesn't exist. However, successful control systems for plants are designed using a combination of readily available tools

eg. PID loops, cascade and feed-forward additions, overrides etc. This industrial approach to control systems synthesis consists of a number of steps. First determine the main flows, secondary flows and recycles, location of surge tanks, product streams with required purity specifications, turndown ratios, complex configurations and the availability of measurements and manipulations. Second the material balance controls are synthesised. Third, the product quality controls are developed for the various units often using plant data to determine the sensitivity of measurements to manipulations. Fourth, controls for secondary flows and temperatures are determined. Fifth, constraint controls and overrides are superimposed on the regulatory structure to maintain operation within feasible boundaries. Finally start-up controls are added. Choices at the different stages are made by experience rather than by analytical techniques. If any doubt exists at the end of this procedure it is resolved by dynamic simulation. This approach is possible because the situations where a particular control tool works are readily understood by the designer. Rule-based programming can turn this apparently fuzzy synthesis method into a usable CAD tool. The maturity of the technology means that this approach can produce useful programs now as long as the appropriate knowledge is available in an expert system.

5.1 Expert Systems

These are often also called Knowledge Based Systems (KBS) and the techniques used to write them rule-based programming. They are computer programs which use models of human reasoning processes in solving problems rather than the traditional algorithms. The typical expert system has 3 parts (fig. 3);

i) A knowledge base that contains the necessary knowledge to solve a problem (unchanged by inference). The commonest type is a collection of rules.

ii) Global data base that contains the facts about the problem to be solved (updated by input and altered during inference)

iii) Control structure that interfaces with the user and finds the problem solution using the knowledge available and the facts in the global database. In a rule based system this step uses backward or forward chaining

Each of these basic components can be expanded in more detail.

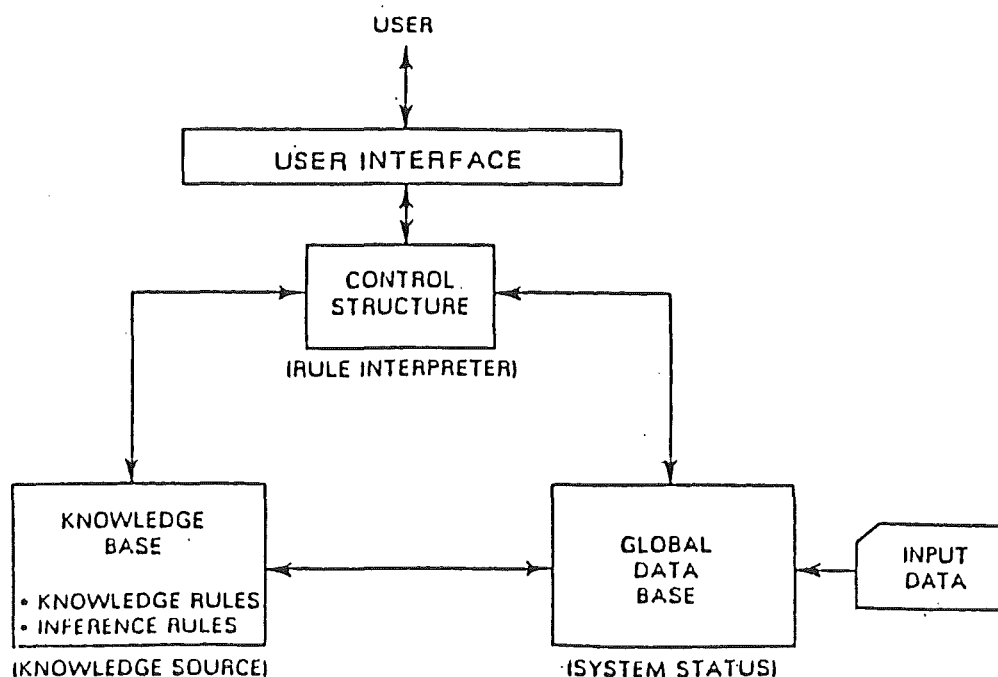


Fig. 3 Typical Expert System Structure

5.1.1 Knowledge and Data Representation

There are two common kinds of knowledge representation in expert systems - "rules" and "frames" and one common data representation type - the "Object-Attribute-Value triple". The statement "Jill has red hair" can be formulated as object "Jill" has attribute "hair-colour" with value "red". In a predicate logic representation this becomes "Jill (hair_colour, red)". The predicate name is the object, the first argument the attribute and the second the value. PROLOG uses predicate logic as the basic statement form.

Rule-based systems use facts and rules to represent knowledge about an area of interest. The O-A-V triples represent the facts. The rules have conclusions that can be drawn if the right facts exist. They take the form "IF antecedent THEN Consequent" where the antecedent must be satisfied by existing facts for the consequent to be true. These are known as IF_THEN or production rules. The domain knowledge will often fall easily into this form.

The facts and rules work well if the knowledge is "flat" ie. there is no hierarchy. If some objects are specific examples of others and inherit some of their properties from parent objects the most convenient knowledge representation type is

the "Frame". A frame is a collection of Attribute-Value pairs that belong to the particular object that gives the frame its name. The attributes in the frame are known as "slots". A slot value can be already defined or ascertained from a procedure or production rule called from the slot.

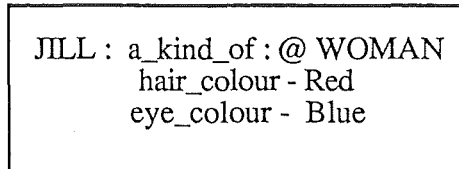


Fig. 4 Frame Knowledge Representation

The a-kind-of link shows that Jill belongs to a class of objects called woman, another frame. All the properties of the woman frame are said to be inherited by the Jill frame. In some systems the frames can inherit slot values from more than one parent frame. This is called multiple inheritance.

There is a further knowledge representation, related to the frame, called the "semantic net". The domain knowledge is organised in a network of nodes and links.

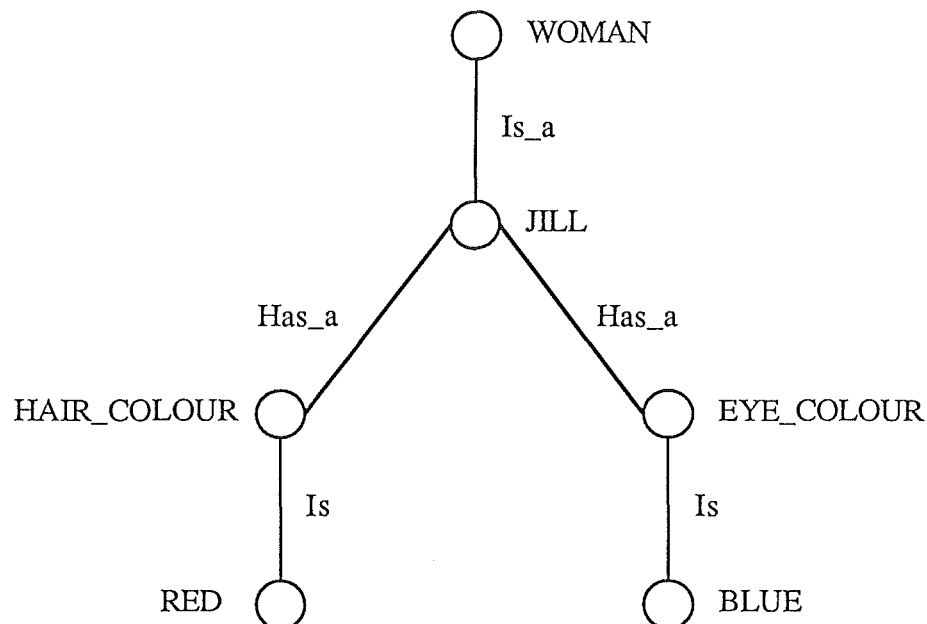


Fig. 5 Semantic Net Knowledge Representation

The is-a link represents inheritance. The has-a link determines a node attribute.

5.1.2 Inference Procedures

The common inference procedures relate to rule based systems. The two types of computer reasoning for rule-based systems are forward and backward chaining.

5.1.2.1 Forward Chaining

Forward chaining is the inference mechanism used by one of the earliest and most well known of expert system tools, OPS5. The procedure begins with a set of facts about a problem stored in the database. The inference engine tries to match rule antecedents from the knowledge base with the facts. If a match is found the conclusion of the rule is added as a new fact to the database and the cycle is repeated. The inference procedure is completed when no more additions can be made to the database. The inference cycle has 3 stages - Match, Select, Execute. The match and execute stages are self explanatory but the select phase is more complex. After the match stage there may be more than one rule which has its antecedents satisfied by the facts. The select mechanism chooses the rule fired during that cycle. The selection criteria may be based on recency of facts, specificity of rule (most antecedents) or order of rules. The conflict resolution strategy may also be based on heuristics for prioritising rules (this represents "meta-knowledge" or rules about rules). If the following rules make up the data base;

1: if A and B then Y

2: if A then D

3: if D then E

4: if D and Y then Z

and the initial facts are A and B. The inference engine will cycle through the rules adding the conclusions of those with their conditions satisfied to the database. The result would be;

cycle 1. rule 1 is satisfied so Y is added to the database.

cycle 2. rule 2 is satisfied so D is added to the database.

cycle 3. rule 3 is satisfied so E is added to the database.

cycle 4, rule 4 is satisfied so Z is added to the database.

The final conclusion is Z and no more information is added in further cycles. Rule order was used for conflict resolution.

5.1.2.2 Backward Chaining

Backward chaining is the reverse of forward chaining. The inference engine starts with an hypothesised goal and tries to prove that it is supported by the facts in the database and the rules in the knowledge base. The first step is for the system to find a rule with a conclusion that matches the current hypothesis. The antecedent of this rule may also be the conclusion of another. The procedure succeeds if the rule chaining ends with a rule that has an antecedent satisfied by the facts. It fails if it reaches a point where the antecedent of the current rule is neither a fact nor the conclusion of another rule. If the procedure fails it should try another hypothesis to see if this can be proven.

If the rules are the same as the previous example and the inference engine is asked to prove if Z is true It would start with rule 4. If Z is true then D and Y must be true. Rule 2 has D as its conclusion and A must therefore be true. As there are no rules with A as their conclusion it must be either a fact in the database or obtainable by asking the user. The other branch of the "proof tree" concerns Y. Rule 1 has Y as its conclusion so A and B must be true. If B was a fact then the backward chaining would be complete and Z would be true. Issues of conflict resolution must also be dealt with in a backward chaining system.

5.1.3 Explanation

A successful expert system should be able to explain its reasoning to the user to allow a judgement on its soundness. The two common types of question that need to be answered are "why" - why does the system need the information, and "How" - how did the system arrive at the current conclusion. The "why" question is easily implemented in backward chaining. An explanation facility can be a trace through a goal tree. Using the above example again if the inference engine reached the point where it was trying to establish A or B by querying the user if, in response, the user asked the machine "Why" ie. why do you want to know that A is true, the machine would reply with the rule " If A then D". If the user persisted and asked "Why" again the system outputs the next level rule "If D and Y then Z". It is then clear how the machine is trying to establish the top level goal.

5.1.4 Knowledge Acquisition

This is the process followed by the "knowledge engineer" to go from the identified problem domain to an organised arrangement suitable for implementation as an expert system. Often the first step in this process is to locate an expert and get an explanation of how a problem is solved. In a series of interviews the knowledge engineer will try to distil the expert's knowledge into a usable form. This means identifying often used rules, the goals and subgoals followed by the expert in solving the problem, identifying the types of questions that the system should ask the user, and perhaps producing a "decision tree" to model the solution process. The process has no fixed procedure and is the bottleneck in the development of an expert system.

"Shallow" and "Deep" knowledge are terms often used to describe the results of the knowledge acquisition process. Shallow knowledge usually comes from interviewing and watching an expert work on examples and breaking this down into rules. The danger with this approach is that the system performance degrades absolutely if faced with a problem outside its scope. Deep knowledge models the real cause of the observed symptoms so that it is generic to the domain. Deep knowledge is more likely to be a causal model than rules, although rules may have some generic application.

5.2 Research on Control Systems Synthesis Using Expert Systems

The most pressing problem in developing an expert system for whole plant control is the structuring and organisation of the knowledge for solution. Although there are many texts available on process control the methods aren't stated in a form readily transferable to an expert system. Bristol (1980) pioneered a method he called "idiomatic control" that promises to be the basis for organising control knowledge. He recognised that most control problems were solved by experts using a set of 20-30 standard "idioms". These are control solutions that work time and time again. Common examples are PID loops, cascade control, feed-forward control but any experienced designer would have variations on these - their own set of idioms. Each of the idioms is appropriate in a particular situation and form generic building blocks to tackle new problems. Using a more complicated example, boiler level three term control is an idiom successful if the contents of a pressurised vessel are on the point of boiling, level control is difficult and material balance calculations are needed to achieve reasonable stability.

This concept was developed by Prassinis, McAvoy and Bristol (1984) who tried to establish idioms and where they were successful by analysing operating control systems. As an example cascade control will correct for flow upsets quickly by cascading a slow controller to a flow loop. In subsequent work by Birky and McAvoy (1988) idioms identified for controlling binary distillation columns were organised using a knowledge representation technique called "Goal-Tree Success-Tree" (GTST) which is shown in fig. 6.

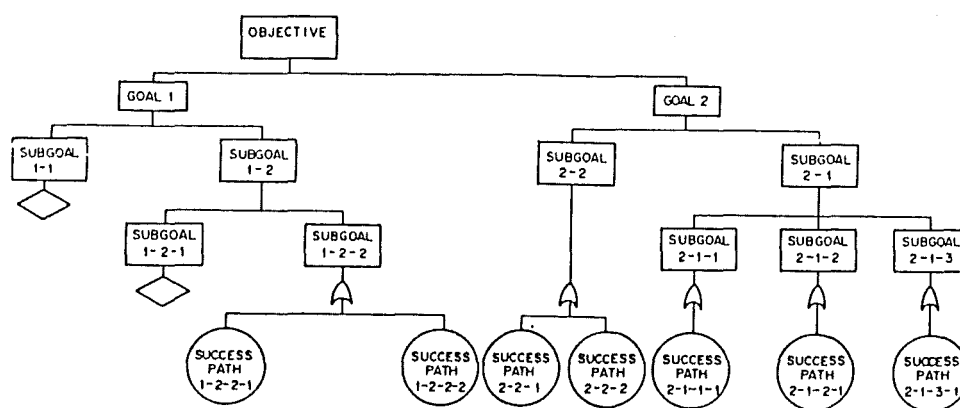


Fig. 6 Goal-Tree Success-Tree Knowledge Representation

The GTST model identifies a top goal which is the primary objective of the expert system. This goal is decomposed into sub-goals that must be satisfied for the the top goal to be true. Each of these subgoals is decomposed into sub-subgoals forming a tree with many levels. At the bottom level in the tree are specific conditions that must be true to satisfy the lowest level sub goals. These conditions identify the "success paths". A partial GTST model for distillation control system synthesis is shown in fig 7. The authors claim that this knowledge representation is easily transferred into a frame based knowledge base. The GTST also lends itself to a backward chaining inference procedure which is the more popular method for shell design.

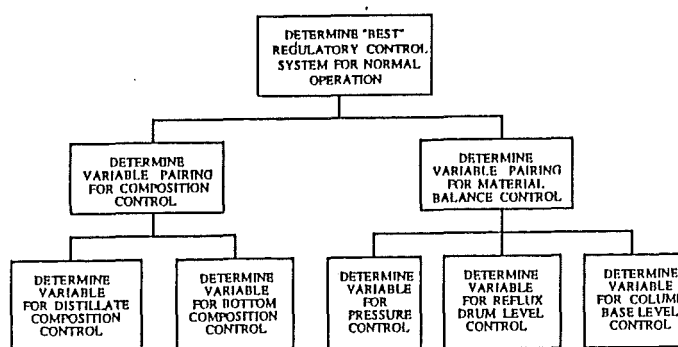


Fig. 7 Goal-Tree Success-Tree representation for distillation control synthesis

Prassinis et al also developed a handy idiom representation scheme which is readily transferable into frames. This provides a useful basis for a graphics interface with the user. This is the obvious communication method because engineers are trained to deal with PID's.

Distillation control system synthesis has attracted interest from other researchers as well. Umeda and Niida (1986) developed an expert system to design the regulatory system for column control. Their synthesis procedure is based around the model shown in fig. 8. Their expert system handles the first 4 stages in this procedure. It is implemented in CHIPS, a production system that uses forward chaining as its primary inference procedure (An advanced version of OPS5). They went on to attempt a generalised system for use in whole plant regulatory control system synthesis (Niida and Umeda, 1986). In this work they used a frame based system (KEE).

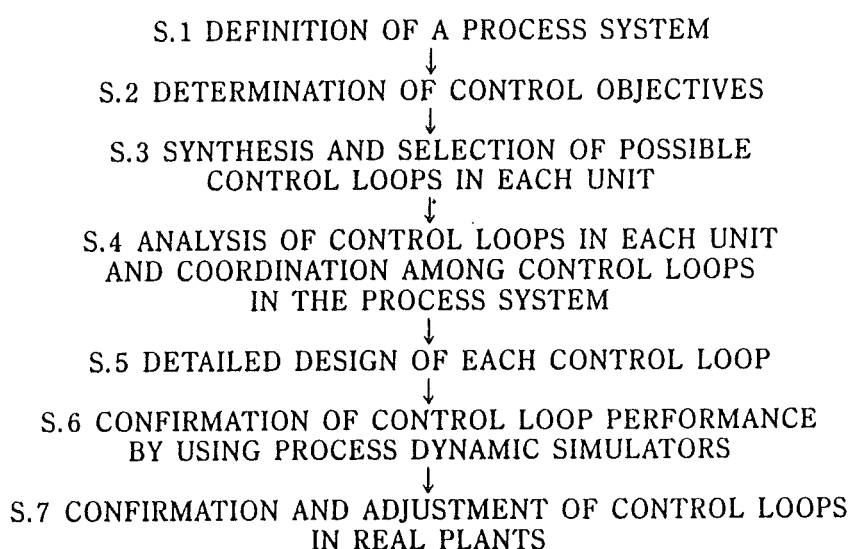


Fig. 8 Control System Synthesis by Umeda and Niida

Shinsky developed an expert system, written in BASICA, a form of BASIC, for a PC that designs distillation control systems (Shinsky, 1986). It relies heavily on the calculation of the relative gain array and a short-cut integrated error procedure to decide the regulatory structure of the column. It has a useful graphics interface with the user but is likely to be a less flexible program than one written in an expert system shell or language.

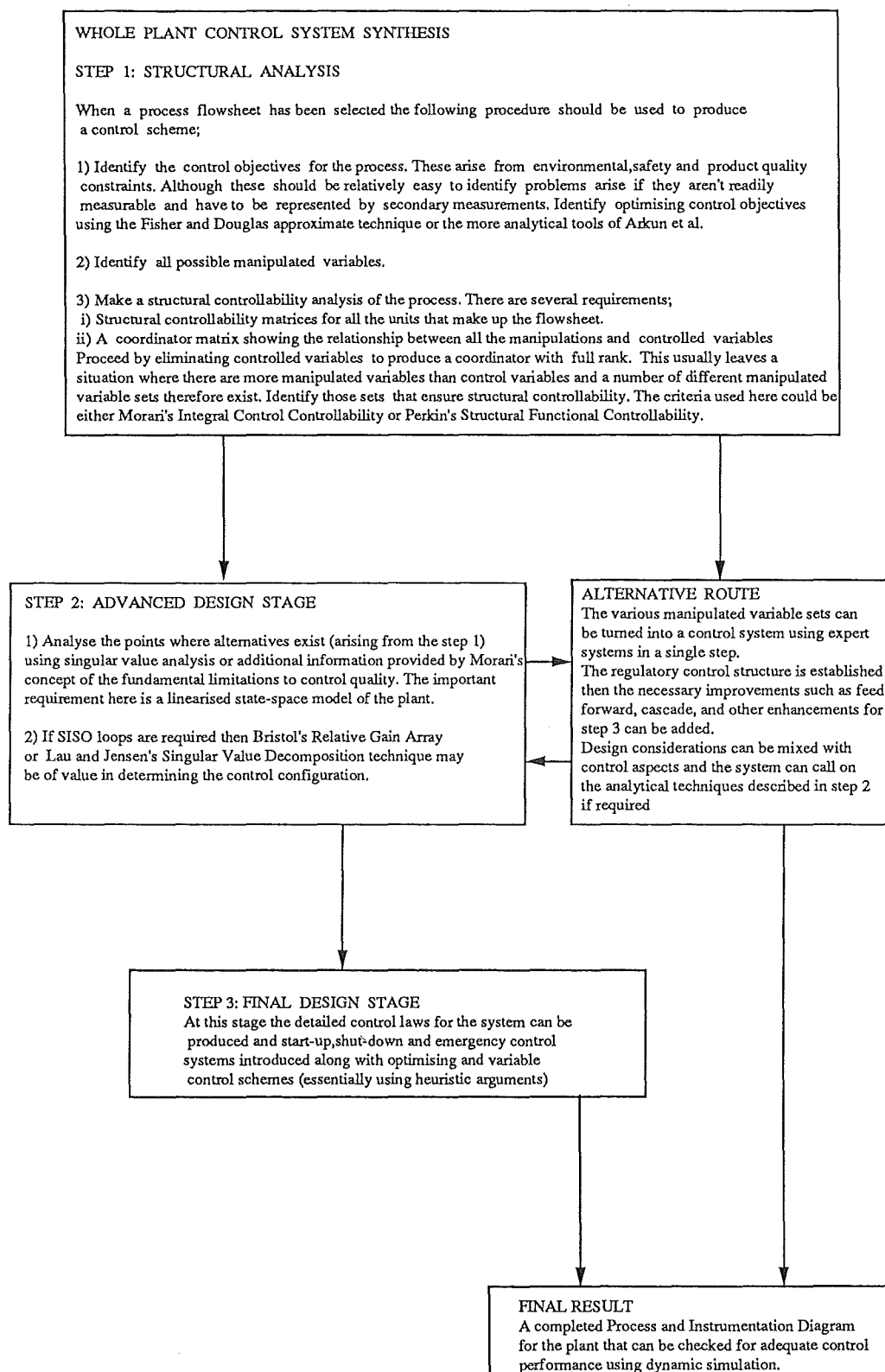


Fig. 9 Piping and Instrumentation Diagram generation by theoretical and heuristic methods

6.0 Conclusion - Piping and Instrumentation Diagram Generation

There is a recurring theme throughout this chapter. In both control systems synthesis and the consideration of control in design there are two approaches, the analytical one which struggles to invent theoretical methods with enough depth to address these complex issues and alternatively heuristic and short-cut approaches already used by engineers to produce workable designs. A diagrammatic summary of these alternatives when used to design a control system for a process flow sheet is illustrated in Fig. 9.

The process begins with the determination of possible manipulated variable sets for the flowsheet using structural controllability analysis and then follows either a theoretical or heuristic path, based around expert systems, to develop a complete PID.

Expert systems may provide a CAD tool that offers a more complete answer in a single step than current theoretical methods. A well designed program would not only use rules but could call upon the "deep knowledge" represented by control theory to answer the design issues that require it. It would also attach importance to design factors that are not immediately concerned with control quality but are none the less vital for a good system design.

In this work, prototype computer programs that serve as the starting point for the development of a complete package for control systems synthesis based on the "alternative route" shown in Fig. 9 are demonstrated. The programs use already developed ideas from structural controllability research to coordinate knowledge based systems that recommend control systems for the unit operations which make up the flowsheet. In one of the unit operations modules, written for distillation column control systems synthesis, Relative Gain Array calculations are used, if required, to improve the synthesis process.

7.0 Nomenclature

d = The vector of output disturbances

G = The process transfer function matrix

G^+ = The transpose of the process transfer function matrix

I = The identity matrix

K = The matrix of controller gains

u = The vector of process inputs

y_s = The vector of steady-state process outputs

$\sigma(\min)$ = The minimum singular value of the process transfer function matrix

$\sigma(\max)$ = The maximum singular value of the process transfer function matrix

γ = The process condition number, $\sigma(\min)/\sigma(\max)$

Chapter 2 - Distillation Control

1.0 Introduction

This chapter begins with a discussion of distillation column control methods and ends with a description of how the bulk of this knowledge was translated into an expert system. The philosophy adopted in this work was that the program recommend likely schemes from the many available but allow the user to make the final decision on which is the most suitable. This differs from other work (Umeda and Niida, 1986) where the program appears to make only one recommendation. Control systems design is too complex an area for heuristic rules alone to decide the design but they are useful to screen out improbable options and reduce any subsequent workload.

The program can handle two product columns that operate with composition control on zero or one of the product streams, and also make recommendations for binary separations requiring dual composition control. The available knowledge could be deepened in two ways. The first would involve adding more recommendations on improvements to the regulatory structure, such as feed forward and constraint control. The second would widen its scope so it could handle more types of column, for example those with sidestream products.

2.0 Distillation Control

There are a large number of schemes suggested to control distillation columns ranging from low cost mass balance alternatives to the more sophisticated and robust dual composition control systems. The selection of the regulatory control configuration decides the control quality of the system. This basic regulatory structure is enhanced using feed forward, constraint and cascade additions to improve the final control. The following schemes, described in the literature, have all operated successfully for particular columns and although not a complete list (it excludes multiproduct columns for instance) it is extensive.

2.1 Mass Balance Control

This is a simple and cheap style of control for a two product column. The feed to the column and one of the product streams must be on flow control and the other product stream on level control to close the mass balance (Fig. 1).

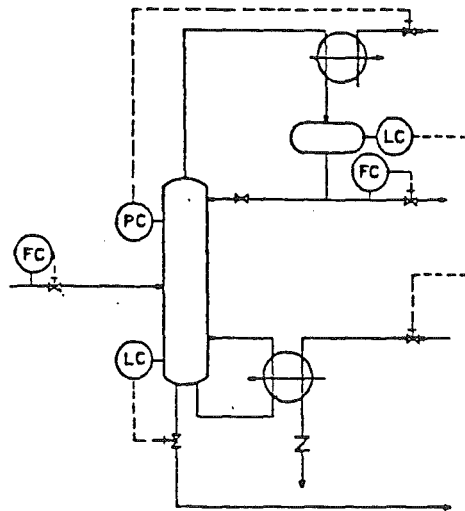


Fig. 1 Mass Balance Control

This type of scheme is a useful solution if the controlled product stream feeds a downstream column because it ensures a steady flow. If there are no disturbances expected in the input to the column (ie. the feed flow, composition and enthalpy are all constant) then this scheme is sufficient to ensure that the product stream compositions also remain constant. This is rarely the case in practice and this approach would hardly ever be successful.

2.2 Single Composition Control Schemes

The most common distillation situation occurs when a product with a critical composition specification is separated from another of a less important composition. The significant product is composition controlled while the other is allowed to vary. There are a number of references that discuss the possibilities for single composition control. This summary (Table 1) is taken from McCune and Gallier (1973) with the following alterations to nomenclature for consistency with this work; distillate flow= D , bottoms flow= B , boilup rate= V , reflux flow= L , boilup rate/feed flow= V/F .

Table 1. Composition Control Alternatives.

Case No.	Overhead Accumulator by	CONTROL		Free	Method for Composition Control
		Reboiler Level by	Composition by		
1	D	L	B	V	Indirect temp *
2	D	L	V	B	V/F
3	D	V	L	B	V/F
4	D	V	B	L	Indirect temp
5	D	B	V	L	Direct temp **
6	D	B	L	V	Direct temp
7	L	D	V	B	V/F
8	L	D	B	V	Indirect temp
9	L	V	D	B	Mixed
10	L	V	B	D	Mixed
11	L	B	D	V	Indirect temp
12	L	B	V	D	V/F
13	B	D	L	V	Direct temp
14	B	D	V	L	Direct temp
15	B	L	D	V	Indirect temp
16	B	L	V	D	V/F
17	B	V	L	D	V/F
18	B	V	D	L	Indirect temp
19	V	D	L	B	V/F
20	V	D	B	L	Indirect temp
21	V	L	B	D	Mixed
22	V	L	D	B	Mixed
23	V	B	D	L	Indirect temp
24	V	B	L	D	V/F

* Indirect temp = Indirect temperature control scheme

** Direct temp = Direct temperature control scheme

There are 24 possible schemes but most can be excluded by one of the following arguments;

1) A loop in the scheme has a large lag or deadtime associated with it, such as controlling reboiler level with reflux flow.

2) A scheme uses boilup rate/feed (V/F) variation to control composition. This control method changes the internal vapour and liquid flows in the column to alter composition without affecting product flows (Fig. 2). A large increase in vapour flow is required to make any significant impact on product composition compared with relatively small changes in distillate flow/ feed or bottoms flow/ feed ratio (typical of the temperature control schemes).

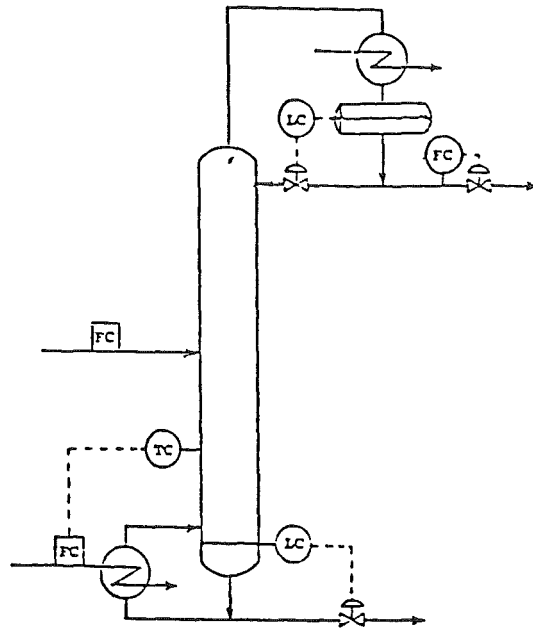


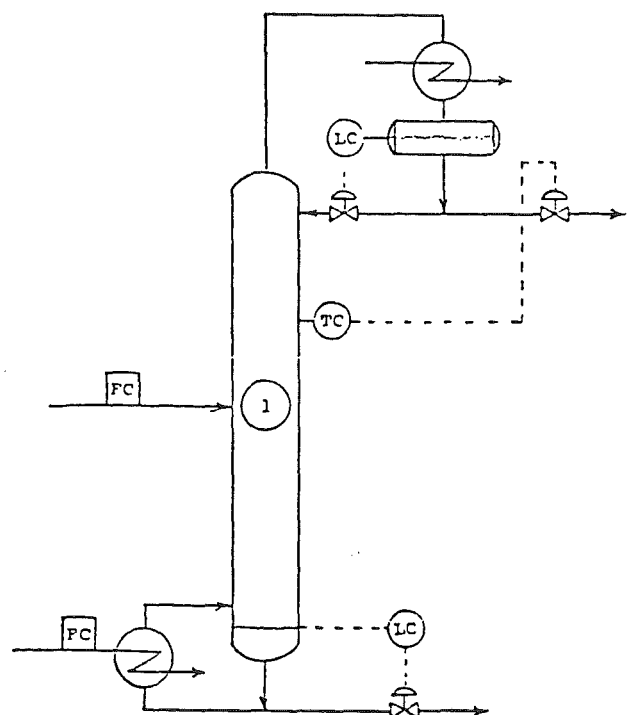
Fig. 2 Control of composition by varying boilup rate / feed flow

3) A scheme presents mass balance difficulties by not having at least one product stream on level control. These are options 9, 10, 21, 22 in Table 1, identified as "mixed"

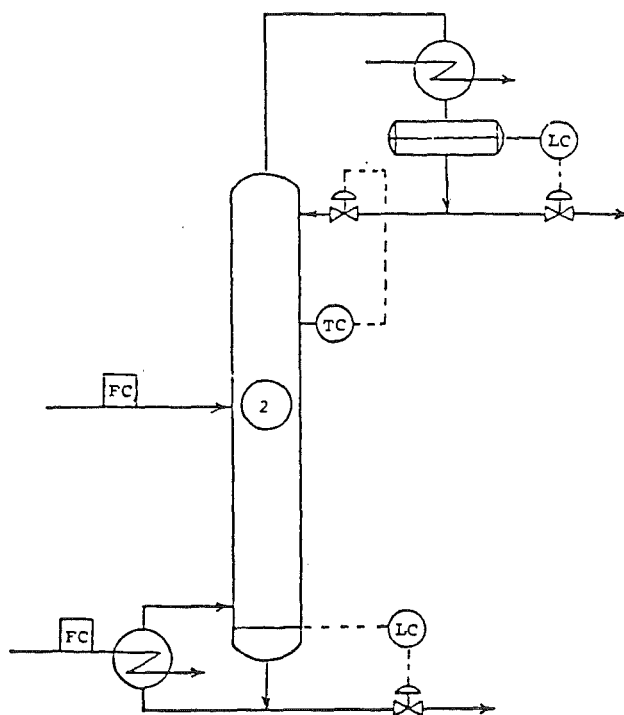
These arguments narrow the choices down to four widely accepted methods that fall in one of two categories, either "direct" or "indirect" temperature control schemes (fig 3). Within these two subgroups dynamic simulations (McCune and Gallier, 1973) demonstrated the superiority of the indirect over the direct schemes when handling upsets in condenser duty. This is explainable because the indirect schemes have automatic reflux control ie. as the reflux is subcooled, top tray vapour flow reduces which in turn reduces external reflux through the level control loop. The internal reflux therefore remains constant and doesn't upset tray compositions. The responses of system 1 to upsets in energy or material balance (variations in feed condition) were also found to be the best. This scheme is recommended by other authors. Shinskey (1977), for example, is strongly in favour of indirect temperature control schemes, which he calls "direct material balance" schemes because of their insensitivity to enthalpy upsets. He goes further to suggest that best control is afforded by using the smallest of the two product flows to control composition as this;

1) Reduces the absolute error in the material balance to the error in the smallest stream under flow control.

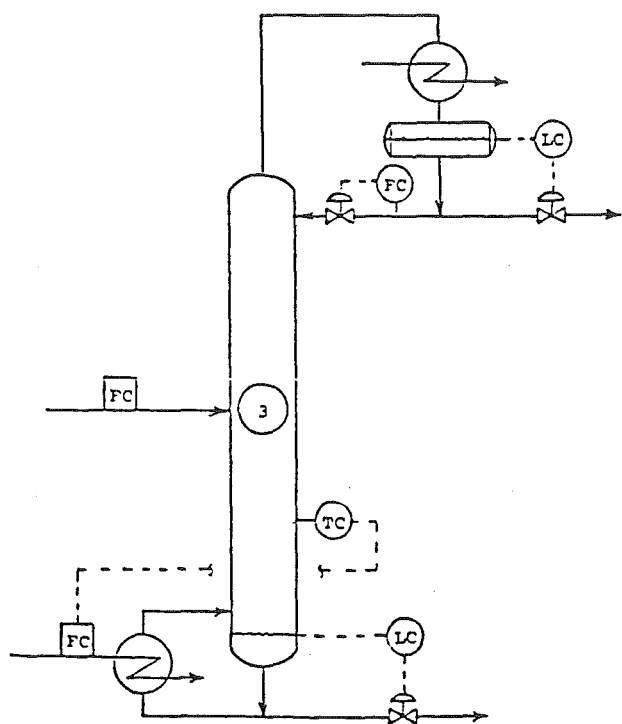
2) Reduces the sensitivity of the control system to feed upsets.



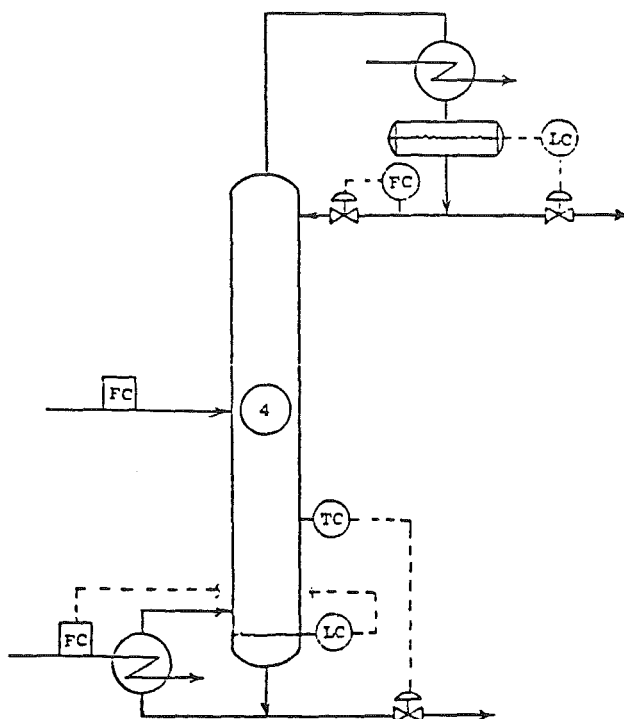
Indirect top temperature control



Direct top temperature control



Direct base temperature control



Indirect base temperature control

Fig. 3 Indirect and Direct temperature control methods

This advantage is considered sufficiently important that he recommends a control structure which uses heat input to control reflux accumulator level and manipulates distillate rate to control bottoms composition when $D/F \ll B/F$. The accumulator level/heat input loop is not normally used by other designers because of the lag in response.

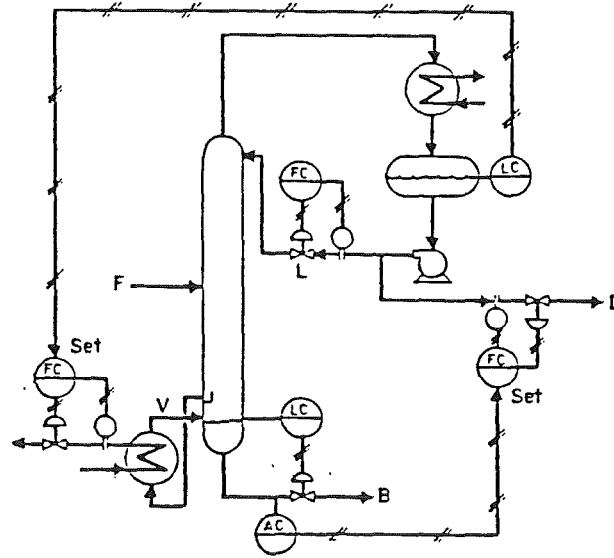


Fig. 4 A scheme recommended by Shinskey (1977) for base composition control when distillate is the smaller product flow

Rademaker, Rijnsdorp & Maarleveld (1975) discuss indirect and direct temperature control. They listed criteria for choosing between the two different classes of schemes summarised as follows;

1) Temperature control on an indirect scheme is always slower than a direct scheme because the level control loop on the accumulator causes a delay before the change in external material balance affects column internal flows. A feed-forward addition to overcome this problem is described by Shinskey (1984), (Fig 5).

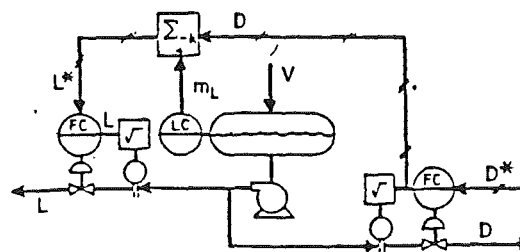


Fig. 5 Removal of Accumulator Lag (Shinskey, 1984)

The output to the reflux control valve is determined as the difference between the level controller output and the measured distillate flow.

2) The indirect temperature control scheme is logically favoured when distillate flow is too small to control accumulator level or bottoms level is too small to control column level.

3) In an indirect scheme the temperature controller introduces variation into the product flow which is undesirable if the stream feeds another unit operation requiring a steady feed.

4) The indirect scheme is more resistant to upsets in energy balance but contrary to the findings of other authors it is claimed that a direct scheme copes better with feed disturbances (this statement isn't supported by any evidence such as dynamic simulation).

5) In some cases there may be more significant interaction between level control using bottoms flow and temperature control using heat input than when the loops are reversed (this is a disadvantage of indirect base composition control).

6) Indirect temperature control schemes can easily be converted to mass balance schemes especially if the temperature controller acts as the primary loop in a cascade configuration onto a product flow controller.

All the temperature control schemes function regardless of whether the composition is inferred from temperature or an actual analyser measurement is used. Accurate single composition control will keep both products on specification as long as there are no significant disturbances in feed flow and composition. In a case where accurate control of both products is important and disturbances upset operation the further complication of dual composition control is justified.

2.3 Dual Composition Control

The best configuration for dual composition control causes least interaction between the two composition loops. Shinskey (1984) details a method and selection criteria for this based on the calculation of the relative gain.

2.3.1 Interaction and Relative Gain

The most widely used measure of loop interaction is the relative gain which is the ratio between two possible open loop gains between a manipulated variable (MV) and controlled variable (CV). The first is the steady state gain between MV and CV with the other loops in the system inoperative so there are no other prospective MV

changes. This is divided by the steady state gain obtained with the other controlled variables constant (all other loops closed).

The relative gain array is a matrix of all possible relative gains between the manipulated and controlled variables in a system and it has the property that the sum of all the elements in each row and column is unity. A two by two array therefore needs calculation of only one of the four elements to complete the entire matrix.

The theory (after Stephanopoulos, 1984) on the use of the relative gain (Λ) as a guide to pairing variables is;

i) If $\Lambda=0$ then the manipulated variable has no direct effect on the controlled variable and this doesn't represent a useful pairing.

ii) If $\Lambda= 1.0$ then the loop is completely decoupled from the others in the system. This is the ideal pairing.

iii) If $0 < \Lambda < 1.0$, the gain increases when other loops are closed, and the smaller the Λ value the more significant interaction becomes and the less suitable the pairing.

iv) If Λ is < 0 then the loop gain changes sign when other loops are opened or closed. This leads to instability.

v) If $\Lambda > 1.0$, the gain reduces when other loops are closed, interaction reduces effectiveness.

2.3.2 Interaction and Dual Composition Control

Level and pressure control loops usually act much faster than composition loops. This means that only interaction between the composition loops needs to be taken into account. The problem is therefore simplified to calculating a number of 2 x 2 relative gain arrays.

In Shinskey's approach the alternative manipulated variables are the distillate flow (D), the vapour flow (V), the reflux flow (L), two independent ratios; reflux flow/bottoms flow (L/B) and boilup rate/bottoms flow (V/B) and the separation factor (S), defined by the equation;

$$S = \frac{y(1-x)}{x(1-y)} = \left(\frac{\alpha}{\sqrt{1 + \frac{D}{Lz}}} \right)^{nE}$$

the second part of the equation demonstrates that separation factor is fixed by controlling distillate flow/reflux flow (D/L). As the ratios reflux flow/distillate flow (L/D), reflux flow/boilup rate (L/V) and distillate flow/boilup rate (D/V) are all dependent on D/L they are also options for separation control.

This makes a total of six independent manipulated variables available and control of any two will fix the end compositions of the column. Therefore there are 15 relative gain arrays. As these are 2×2 arrays they are completely characterised by calculating or measuring the first element in the array. A worksheet of these values can be evaluated for a particular column and used as a reference to make a selection.

		Control y with →				
		D	L	L/D	D/V	D/L
Control x with ↓	D		Λ_{LD}			
	B				Λ_{SD}	
	V	Λ_{DV}	Λ_{LV}		Λ_{SV}	
	B/L	$\Lambda_{DL/B}$	$\Lambda_{LL/B}$		$\Lambda_{SL/B}$	
	V/B	$\Lambda_{DV/B}$	$\Lambda_{LV/B}$		$\Lambda_{SV/B}$	
	V/D	Λ_{DS}				
	V/L					

Fig. 6 Relative Gain Worksheet

There are gaps in the worksheet where pairings are unlikely or impossible. For example, D can't be used to control both x and y (this option is shown shaded in the worksheet) and neither can separation factor (the entire right hand bottom corner of the worksheet is missing because of this).

In applying relative gain theory to column control if the relative gain is greater than 1 and in the range 1 to 5 or less than 1 and in the range 0.9 to 1.0 the control configuration satisfies this design criterion. This appears to be based on Shinskey's experience rather than any more rigorous evidence.

2.3.3 Selection Criteria

Shinskey suggests some other factors that should be taken into account along with interaction when configuring the control system;

i) The smallest flow should be manipulated to control composition because this reduces the error in column material balance to the error in the flow of the smaller product flow.

ii) The accuracy of the material balance is also important when considering which ratio to use when controlling the separation factor. The smallest ratio should be used eg. if $D < L$ manipulate D/L (or, even better still, D/V as $V=L+D$) and use L for accumulator level control. The ratio would be L/D if $L < D$.

Considering these factors and the relative gain worksheet there are several commonly acceptable configurations

1) The top composition controlled by separation factor and the bottom composition by boilup rate/ bottoms flow (V/B). This is known as the SV/B configuration (fig 7) and should be used when $\Lambda_{SV/B}$ (the relative gain for the SV/B scheme) is in the range 1-5 and distillate, D , is smaller than B . This represents the closest thing to a universal solution to dual composition control as it has a fast response and the smallest relative gain of the group of options that have relative gains greater than 1.

In the SV/B scheme the top composition controller outputs the required D/V ratio which is converted to a set point for a distillate flow controller by multiplying this signal by the level controller output ($V = L + D$). Changes in distillate flow are transmitted to reflux flow by subtracting its measurement from the output of the level controller ($L = V - D$) and using this as a setpoint for a reflux flow controller. The bottom composition controller outputs the required V/B ratio which is converted to a setpoint for a boilup flow controller by multiplying by the bottoms rate B .

This scheme is superior to the simpler SV (fig 7) configuration that has been used successfully in industry (Ryskamp, 1980) because it has improved interaction characteristics. The SV scheme in Fig 7 uses D/L as the ratio to control separation factor. This is slower than the D/V choice because any change in distillate rate is only transmitted to the column when the accumulator level loop alters reflux flow. The D/V option immediately affects reflux flow through the summing junction when distillate flow is changed by the composition controller.

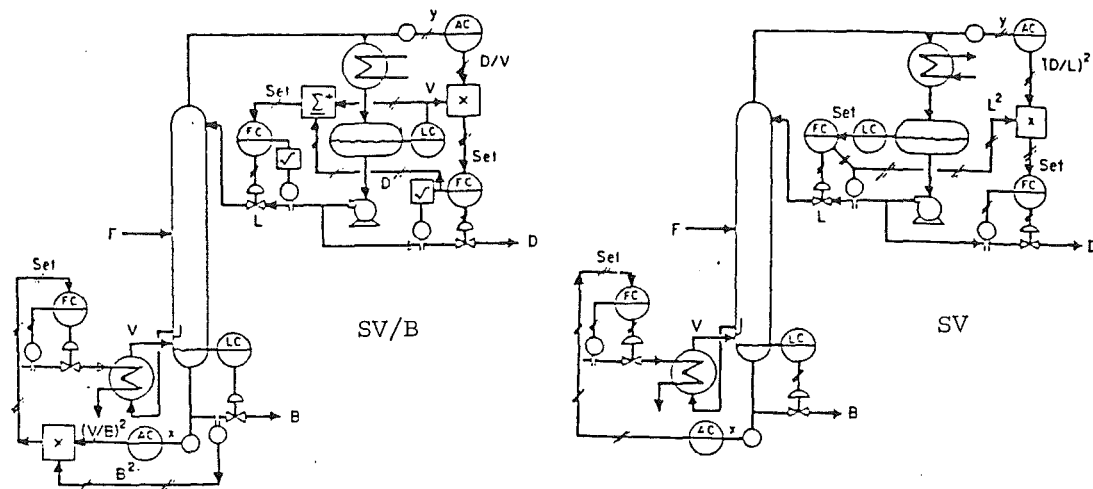


Fig. 7 Top composition controlled by separation factor and bottom composition by V/B (SV/B) or by V (SV)

2) The top composition controlled by distillate flow and the bottom composition by boilup rate/bottoms flow (V/B). This is known as the DV/B scheme (Fig. 8) and is applicable when $\Lambda_{SV/b} > 5$ and $\Lambda_{DV/b}$ (the relative gain for the DV/B scheme) is in the range 0.9 - 1.0 and the distillate is the smallest flow.

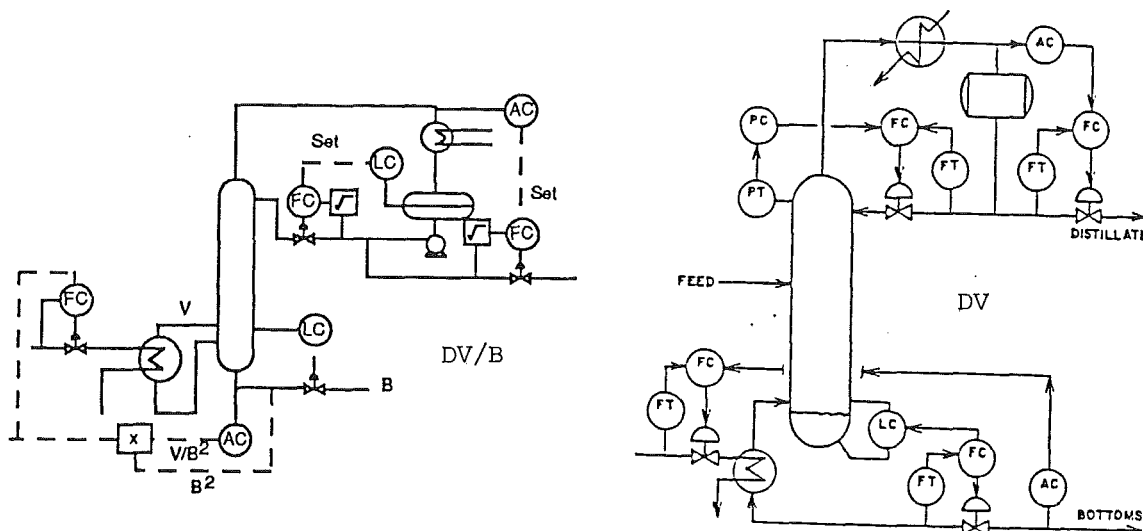


Fig. 8 Top composition controlled by distillate flow and bottom composition by V/B (DV/B) or by V (DV)

The simpler DV scheme (Fig. 8) has worked successfully on industrial columns with high reflux ratios ($L/D > 5$) but the V/B substitution in place of V

usually improves the relative gain. The major problems with this simpler scheme is a relatively sluggish response especially in the bottom loop and a failure to control if the top loop is left open. If this happens the heat input valve tends to saturate if there is a feed composition upset, such as an increase in the light components, because the material balance is fixed. The lights accumulate on the trays and finally collect in the base of the column. The boilup rate increases to maintain bottom composition but as the distillate remains constant the light components aren't easily removed and the heat input valve will be forced to open fully to maintain control.

3) Top composition controlled by separation factor and bottom composition by distillate. This is known as the SD scheme (Fig. 9) and is applicable when $\Lambda_{SV/b} > 5$, $D < B$ and Λ_{SD} (the relative gain for the SD scheme) is in the range 0.9 - 1.0.

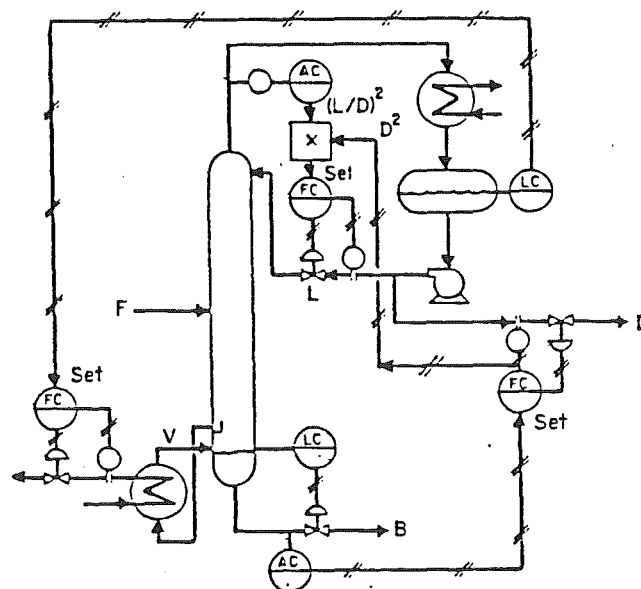


Fig. 9 Top composition controlled by separation factor and bottom composition by distillate rate (SD)

This scheme includes the pairing of accumulator level with boil-up which isn't widely accepted because of the lags inherent in the column. The compromise is made here because the pairing of distillate, the smaller product flow, and bottoms composition improves the accuracy of the column material balance.

4) The top composition controlled by separation factor and the bottom composition by bottoms flowrate. This is known as the SB scheme (Fig. 10) and is applicable when $\Lambda_{SV/b} > 5$, $\Lambda_{SD} = \Lambda_{SB}$ (the relative gain for the SB scheme) is in the range 0.9 - 1.0 and the bottoms flow is the smaller product flow.

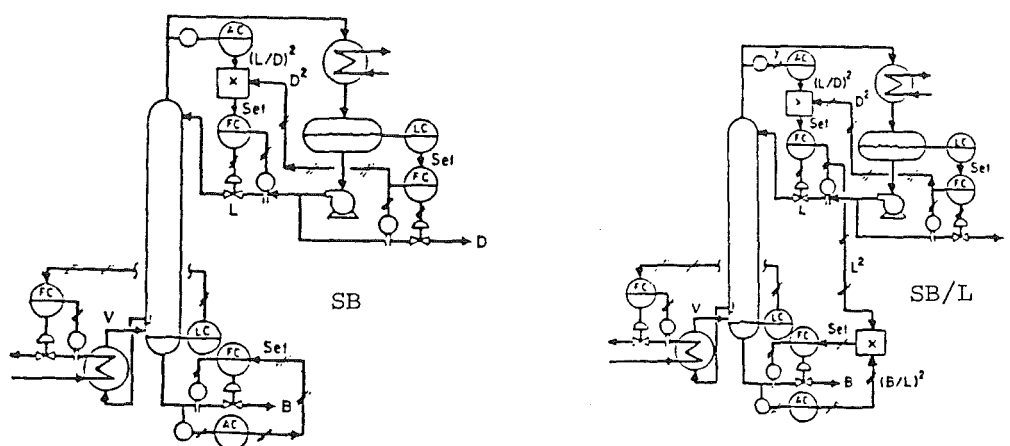


Fig. 10 Top composition controlled by separation factor and bottom composition by bottoms flow (SB) or the ratio B/L (SB/L)

If $\Lambda_{SV/b}$ is in the range 1-5 but the other factors still apply then a better solution is to control bottom composition using B/L as this has a relative gain only slightly more than the SV/B scheme and preserves accuracy in the bottom loop. A simpler version (LB) is also possible but this choice compromises relative gain for a slightly cheaper installation.

A number of other authors have supported this approach. Gordon (1987) used relative gains in an identical fashion to select control systems. Takamatsu et al (1987) recommended a scheme very similar to the SV/B system to minimise interaction. Their scheme has top composition paired with L/V and bottom composition paired with $V/(L+B)$. Waller et al (1988) carried out experiments on an actual column to compare the traditional schemes (LV and DV) with the recommended new methods, SV and SV/B . They concluded that the LV, and the SV schemes handled feed composition upsets best while the SV/B scheme was better for feed flowrate disturbances. The better schemes all had similar control characteristics but the DV scheme was considerably worse in all cases. The LV scheme has its best relative gain at low reflux ratios ($L/D < 1.0$) and this degrades as the ratio increases. The experiments were done using a column with a low reflux ratio so the reasonable performance of the LV scheme therefore isn't unexpected. The DV scheme has significant interaction under these conditions and is unsuitable although it has proved successful if the reflux ratio is high.

Ryskamp (1980) reported successful implementation of an SV scheme and a refinery in Australia recently presented results on double-ended temperature control of a column using the same approach (Rowney et al, 1987).

Two schemes, top composition controlled by reflux and bottom composition by boilup (LV, Fig. 11) and the DV configuration (Fig. 8), have traditionally been used. Although simpler, in many cases these approaches have poorer relative gains and almost always slower dynamics than the schemes discussed (Shinskey, 1984).

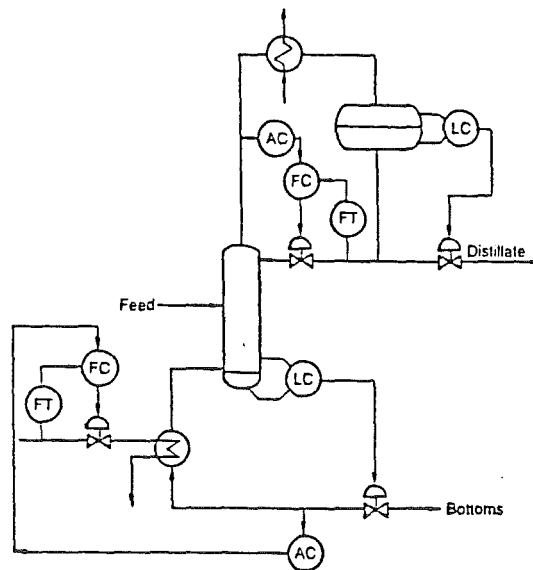


Fig 11 Top composition controlled by reflux rate and bottom composition by boilup (LV)

2.4 Improvements to The Regulatory Structure

Additions to the basic regulatory structure for the column that improve control quality and avoid operational constraints are discussed in this section.

2.4.1 Feedforward Control

There are feedforward schemes recommended in the literature for both single and dual composition controlled columns.

2.4.1.1 Single Composition Control

Feedforward control is added to a column to improve product composition control when the system is upset by changes in feed flow or composition. If a column is fed under level control from an upstream column the feed may develop a regular sinusoidal variation and feedforward control can stabilise column operation. It isn't

normally implemented on its own because the feedforward models aren't accurate enough to predict response exactly but is used to enhance feedback loop performance. There are two types of feed forward configuration suggested by Shinskey (Liptak, 1985). The first operates by maintaining separation in the column by adjusting heat input in ratio to feed and controls distillate composition by varying distillate rate proportionally to feed (Fig.12).

Feedback from an analyser on the distillate stream can be used to alter the calculated value. The other variation is used for maximum recovery of a single product and involves holding boil-up at its maximum rate and varying distillate to maintain its composition at a constant value. Here separation varies and distillate flow must be approximated by a parabolic function; $D=mF - bF^2$. This is implemented as shown in Fig. 13.

Feed forward control usually requires dynamic compensation to ensure that control action is made at the correct time thus compensating for column lags.

2.4.1.2 Dual Composition Control

Some of the suggested dual composition control schemes can be improved by feedforward control of feedrate upsets. If a DV configuration is selected then a similar addition to the one applied to the single composition control scheme can be used (fig. 14). The top composition controller outputs the required D/F value which is multiplied by a dynamically compensated feed flowrate signal to provide the setpoint for the distillate flow controller. The base composition controller outputs the required heat input/feed ratio (sets V/F) which is multiplied by the feed flowrate signal to provide a set point for the heat input controller (usually a flow controller on the reboiler heating stream).

If an SD scheme is used then only one feedforward addition is needed. The base composition controller outputs required D/F which, after multiplication by dynamically compensated feed flowrate, sets a distillate flow controller.

The SV scheme similarly only requires feedforward compensation on the base composition controller. This would be exactly the same as used for the DV case (fig. 14). The more complicated version, SV/B, isn't improved by feedforward control. If a liquid feed flow change occurs it is transmitted via the trays to the bottom of the column where bottoms flow is changed by the level controller. This causes an immediate change in the boilup. When this affects the reflux accumulator level reflux

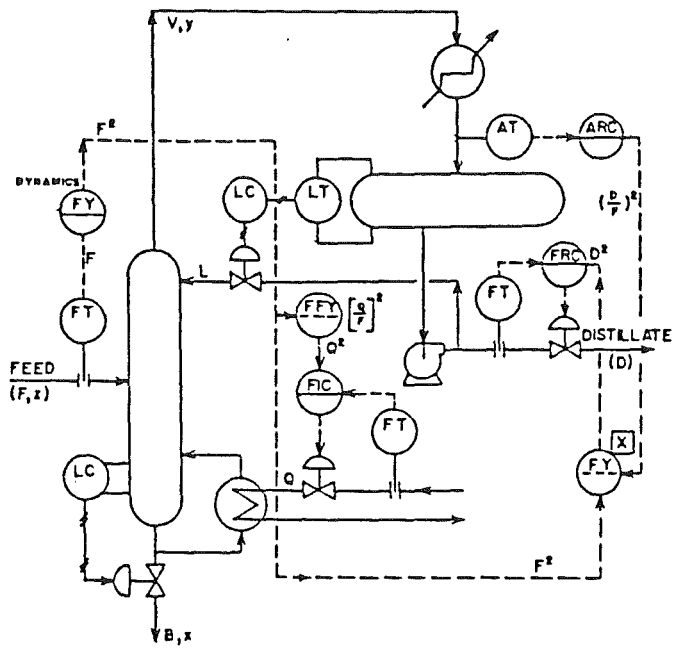


Fig. 12 Constant separation

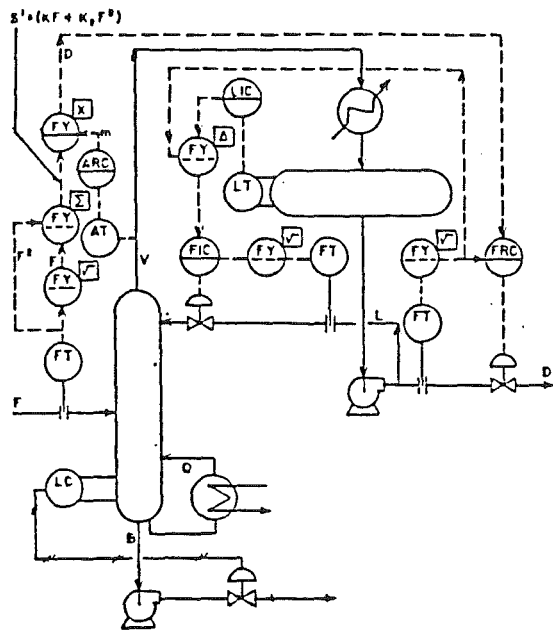


Fig. 13 Maximum Recovery

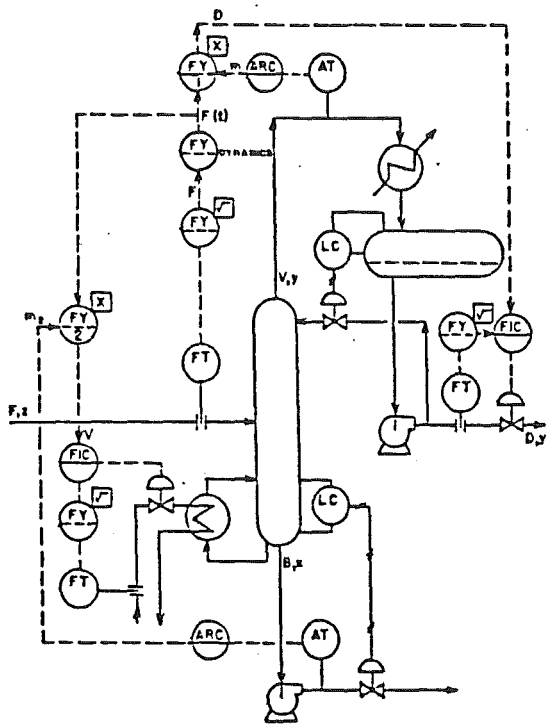


Fig. 14 Feedforward enhanced dual composition control

and distillate flow will also change appropriately. The necessary alterations in V, B, D and L are all made at the right time to maintain product compositions without any feedforward additions.

2.4.2 Cascade Control

Primary level or composition control loops are often cascaded onto a secondary flow control loop to smooth out any variations in flow. The figures 7 - 14 all show instances of this. For example, in Fig. 14 the calculated distillate flow setpoint is cascaded onto the distillate flow controller. This inner or secondary loop handles any fluctuations introduced into the distillate flow and therefore improves the overall performance of the composition controller.

2.4.3. Constraint Control

One of the most significant constraints in the operation of a column is the tray operating window. Operation must always occur with a boil-up rate that causes sufficient vapour flow in the tower to prevent "weeping" but not enough to "flood" the column. A method to control flooding is to use a pressure drop controller, acting from pressure drop measurements over the trays, to override the composition or level control signal via a low signal selector if the DP gets close to the flooding limit.

2.4.4. Internal Reflux Control

If the enthalpy of the external reflux stream varies then internal reflux flow and tray compositions will also vary, even though the external reflux flow remains constant. Internal reflux control is a solution to this problem (Liptak, 1985 and Shinskey, 1977 and 1984, Harriott, 1964). The reflux flowrate is corrected according to its degree of subcooling. If the reflux becomes significantly subcooled below the vapour temperature then external reflux will be reduced (Fig. 15)

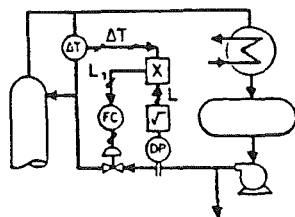


Fig. 15 Internal Reflux Control

2.5 Distillation Column Pressure Control

Many columns operate at atmospheric pressure and for these no pressure control is necessary. However for the columns that need it there are a significant number of column pressure control methods presented in the literature. Although it is possible to control pressure using boil-up this is rarely used in practice and the available methods manipulate heat removal from the column. Chin (1979) discussed 21 different methods for pressure control depending on the type of condensing equipment available. The common methods are shown in table 2.

Table 2. Pressure Control Methods

METHOD	CIRCUMSTANCES USED	OPERATION	COMMENTS
1) PC valve in the vapour product line exit the reflux drum	vapour product sent to a lower pressure, partial condenser, air or water cooled	Directly alters column pressure by altering the removal rate of vapour from the column	The method is direct and easy to understand. The speed of response depends on the size of the product stream as does the valve size.
2) PC valve in the vapour product compressor recycle line (vapour product drawn off the reflux drum by a compressor)	vapour product sent to a higher pressure, partial condenser, air/water cooled	The product flow and therefore pressure can be regulated using a spillback line from compressor discharge	As the valve is in a recycle line it would be smaller than for method 1.
3) PC valve in the condenser inlet (An equalising line connects column to reflux drum upstream of the control valve)	no vapour product, flooded condenser above reflux drum, drum and column operate at the same pressure, condensate ex condenser enters the drum below the liquid level, air/water cooled	As the valve operates it varies the condenser pressure causing liquid level to rise or fall in the condenser and therefore vary heat transfer area	The valve is in a vapour line and is bigger than in an alternative method (4) which has a similar speed of response.
4) PC valve in the condensate line ex the condenser (reflux drum runs at column pressure)	no vapour product, flooded condenser above reflux drum, drum and column operate at the same pressure, static head provides the driving force across the valve, air/water cooled	As the valve operates it varies the liquid level in the condenser This has the same effect as explained in (3).	The valve is in a liquid line and is therefore smaller than in (3) with a similar response speed. The gravity flow pipework between condenser and drum needs careful design to ensure sufficient valve driving force.

METHOD	CIRCUMSTANCES USED	OPERATION	COMMENTS
5) PC valve in "Hot Gas Bypass" around the condenser	no vapour product, condenser BELOW reflux drum, condensate must be subcooled, bypass between column and reflux drum.	The bypassed hot gas heats the reflux surface altering drum pressure and forcing liquid back to the condenser reducing heat transfer area	This method has failed to work on some installations. Design is empirical and not easily understood but has the design advantage that ground level condensers are inexpensive to install and service.
6) PC valve altering reflux flow (reflux drum flooded)	no vapour product, flooded condenser, flooded reflux drum (or excluded for faster response)	Operates by varying level in the condenser. A similar approach to (3) and (4).	Has the advantage that a level control system isn't required for the reflux drum. Has a similar response speed to other flooded condenser techniques. Faster quality control is possible.
7) PC valve in the condensate line ex the condenser (reflux drum under pressure control using a valve manipulating bypass flow around the condenser)	no vapour product	very similar to (4)	This method has a similar response time to (4). The condenser positioning and the condenser/drum piping design isn't as important because the valve driving force is assured by reflux drum pressure control.
8) PC valve throttling coolant to condenser	no vapour product, not air cooled, top temperature < 50 degrees C if water cooled	The controller varies coolant flow to alter heat transfer coefficient and control pressure	If cooling water is used it may reach overhead line temperature in a throttled condition. If this exceeds 50 degrees C excessive fouling may occur. This approach introduces the lags and non-linearities peculiar to this heat exchanger control method.
9) Alter fan pitch on the fans of an air cooled condenser	no vapour product, air cooled condenser	Heat flux is varied by altering the air flow through the cooler. A similar approach to varying liquid coolant flow.	This method is sensitive to weather condition upsets and is more expensive than (9).
10) Alter louver position in the air flow to or from an air cooled condenser.	no vapour product, air cooled condenser	Achieves variation in the heat flux as in (9) . The louvers vary air flow in much the same way as a control valve in a liquid circuit	Similar comments apply to this approach as to (9) but this is a cheaper solution.

METHOD	CIRCUMSTANCES USED	OPERATION	COMMENTS
11) Throttle total overhead vapour flow from the column	zero/intermittent flow of vapour, air/water cooled condenser	Valve simply throttles the vapour to the condenser. The condensing pressure in the condenser alters to maintain total condensing when the reflux vent is closed or operates at atmospheric pressure if it is open.	Although this method offers fast, tight control the valve would be large and expensive for columns with large overhead vapour flows. If the valve fails closed the the column will rapidly overpressure.
12) Split range of inert gas fed to the reflux drum or non-condensibles gases vented via a vent valve	Intermittent flow of vapour, air/water cooled condenser, inert gas available at higher than column pressure	If the pressure rises the vent valve vents the non-condensibles and if pressure falls the inert gas valve opens to admit inert gas.	This method can be wasteful of inert gas and requires careful design to ensure that it can escape when not required

3.0 Expert System For Distillation Column Control Systems Synthesis

An expert system was written to synthesize regulatory control structures for distillation columns as a prototype for a more comprehensive system. The first version was written using an expert system shell (Millen, 1987) acting in its forward chaining mode. This approach had some drawbacks. There was no easily programmed method within the shell for generating a group of solutions and this tended to force a single answer to any problem. This goes against the philosophy of suggesting a number of likely possibilities to the user. There was also a difficulty in structuring the rulebase to sensibly come to one answer. This had to be solved by adding rules that prevented unwanted conclusions being drawn. As an example, bottoms flow was the first choice to control column level and in order for the program to conclude that heat input should be used in this loop then bottoms flow must either control base composition or be on flow control. These rules add nothing to the knowledge about the problem but must be there to ensure that no more than the required number of control loops are suggested. These rules make the rulebase complex and hard to follow. This inflexibility in the shell was a major reason for using PROLOG, which has a greater capability, in the rest of the work. More details on the forward chaining version and its attendant problems are included in Earl and Williamson (1988).

The PROLOG program works effectively on two product columns with one or two compositions or none included as control objectives. It could be extended to include extra rules to cover columns with sidestreams or multicomponent columns

requiring dual composition control. The program is written as a hierarchy of goals, the natural form for a backward chaining inference engine. The top goal, "form_combinations", requires the successful formation of all possible control combinations for the column. This is achieved when the subgoals shown in fig. 16 are satisfied.

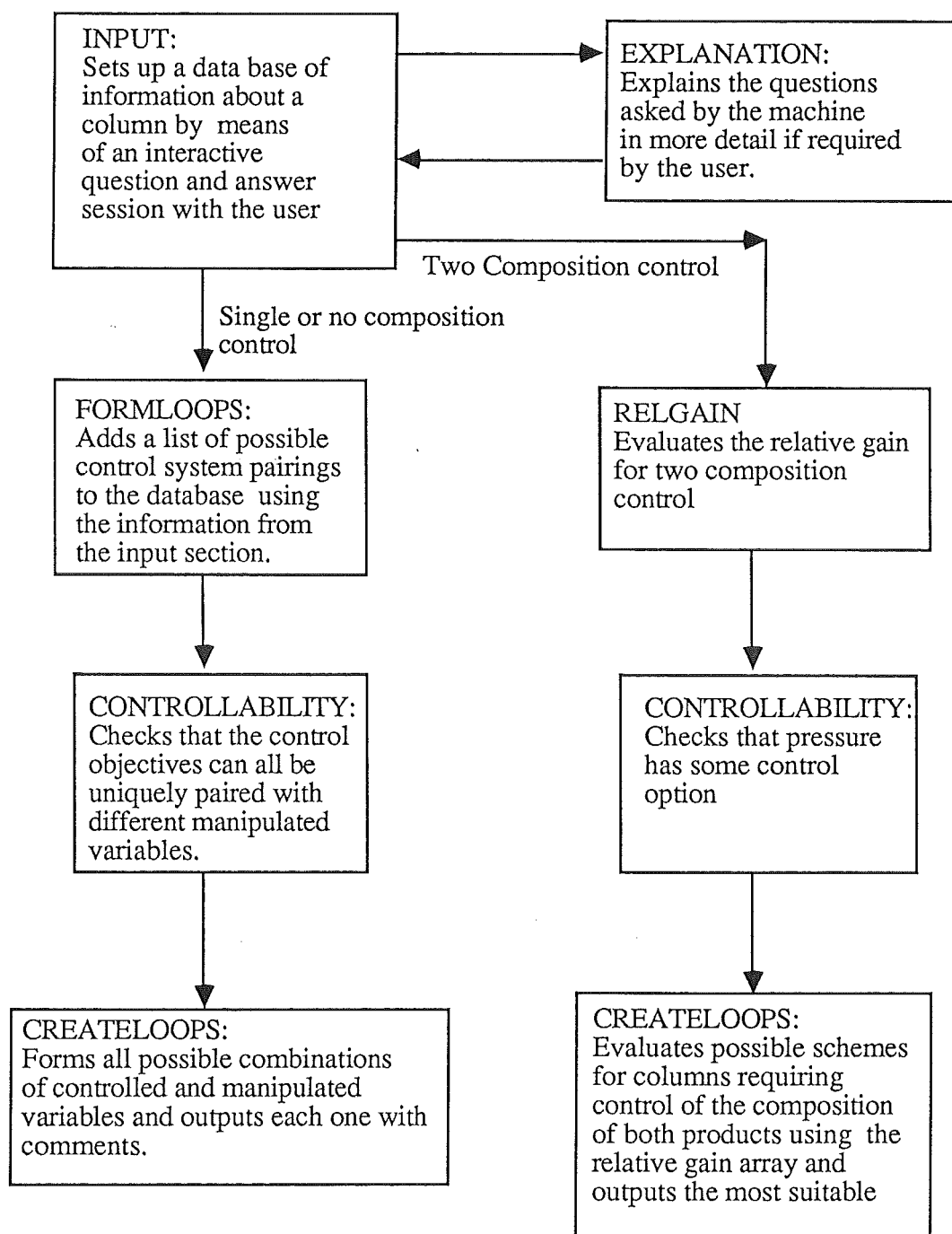


Fig. 16 Operation of the distillation control synthesis expert system. (The subgoals are evaluated in sequence to satisfy the top goal.)

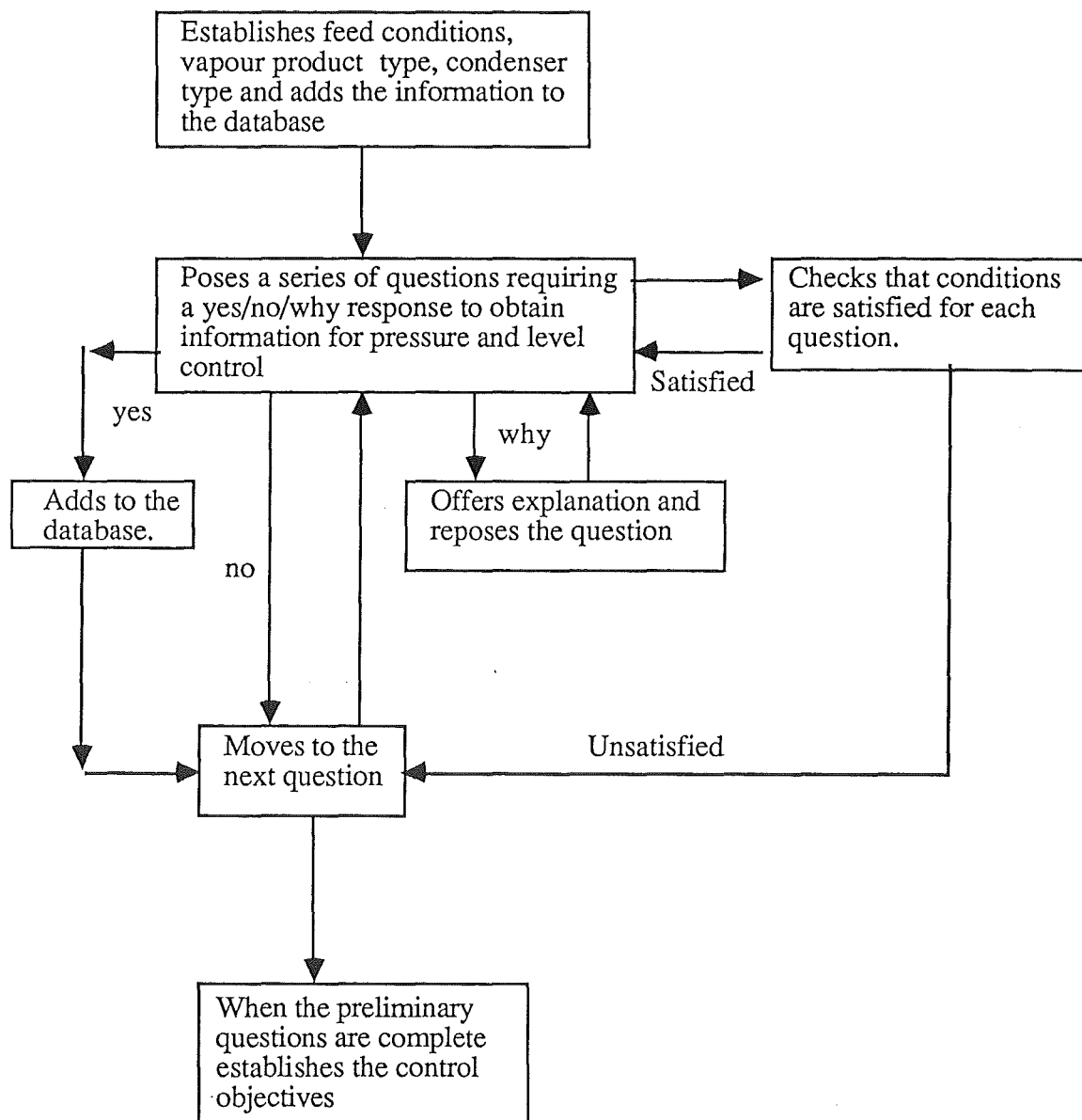


Fig. 17 Flow diagram of the operation of the "Input" subgoal of the expert system

3.1 Input Section

The input section, a block diagram of its operation is shown above in Fig. 17, sets up the database of information for a particular column in an interactive session with the user. The first series of questions establishes the feed condition, the type of vapour product, if any, and the condenser type. The next and larger set of questions is mainly on conditions for pressure and level control and are programmed to display

extra information if required. All of these can be answered with either yes, no or why. The why response initiates a further explanation of the question. As the database builds up some of the questions become unnecessary and are only asked if the correct preconditions exist in the database. The final part of the input section finds the control objectives for the column and places a list of them in the database. The conventional column has up to five control objectives (pressure, reflux accumulator level, column level, top and bottom composition) and five simple manipulated variables (condenser cooling rate, heat input, distillate flow, bottoms flow and reflux flow). The feed is usually not available for control. If a particular manipulated variable is specified as being on flow control (one of the control objectives) it limits the possible control combinations for the column. It is better to underspecify the control objectives and allow the program to recommend more alternatives each with different spare manipulated variables. These spare variables would have to be placed on flow control for completeness but the choice is made by the program.

3.2 Formloops section

This section uses a set of rules to establish the possible control loops for a column. The structure of this section, shown in fig. 18, is very similar to the "Goal-Tree Success-Tree" concept used by Birky and McAvoy (1988). There is a primary goal for the section, the formation of all feasible control loops, and this is satisfied when rules grouped under a number of categories are checked with the database. If a rule "fires" then a success path is found and a candidate loop is added to the database. As an example, one of the rules is "If the condenser is water cooled and the top temperature of the column is less than 50 °C then column pressure can be controlled by condenser cooling rate". The rule prevents cooling water manipulation to control pressure if there is a chance of fouling occurring on heat exchanger tubes. The program sweeps through all the rules during the evaluation of a column.

The program cuts down the number of possibilities by ensuring that unlikely loops are never recommended. The top temperature (used to infer top composition) or reflux accumulator level would never be controlled by bottoms flowrate because of the lags and deadtime in the column so there are no rules to suggest these combinations. All the feasible control loops are written out as a record for the user by forced backtracking as the final subgoal in the section.

The program operates differently if two composition control is specified for a column. Only pressure control loops are specified in this section, before the program branches to the next stage.

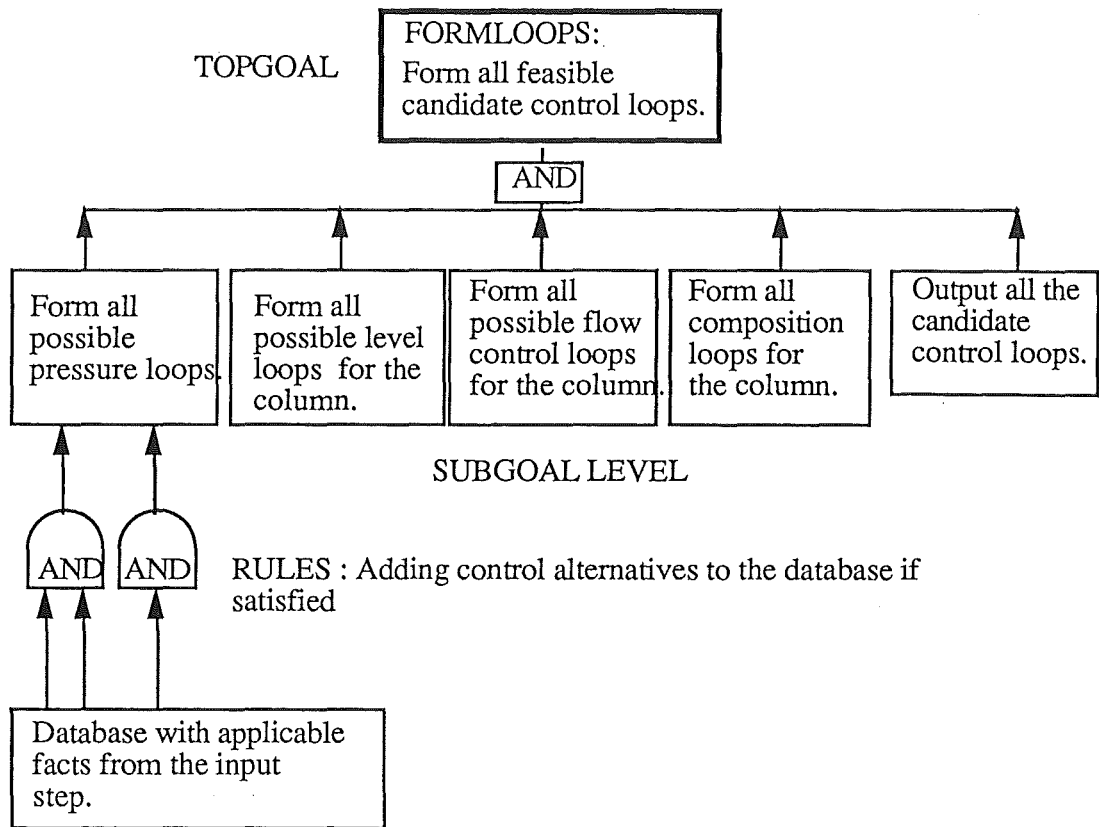


Fig. 18 The structure of the "Formloops" subgoal of the program

3.3 Controllability Section

If single or no composition control is specified this part of the program checks whether a complete regulatory control system can be formed from the available control loops. Its structure is shown in fig. 19. If no loops have been proposed for a control objective or there are two control objectives that share a single manipulated variable then the check fails. However if it can form at least one combination where each control objective is paired with a distinct manipulated variable this condition is satisfied. This is not a complete functional or state controllability check.

Two composition control requires only that a pressure control loop is available and the correct control objectives have been specified. The existence of a pressure control loop is unnecessary if the column operates with a flooded drum because pressure replaces accumulator level as a control objective in the recommended control systems. If the check fails a message is printed out for the user and the program won't continue.

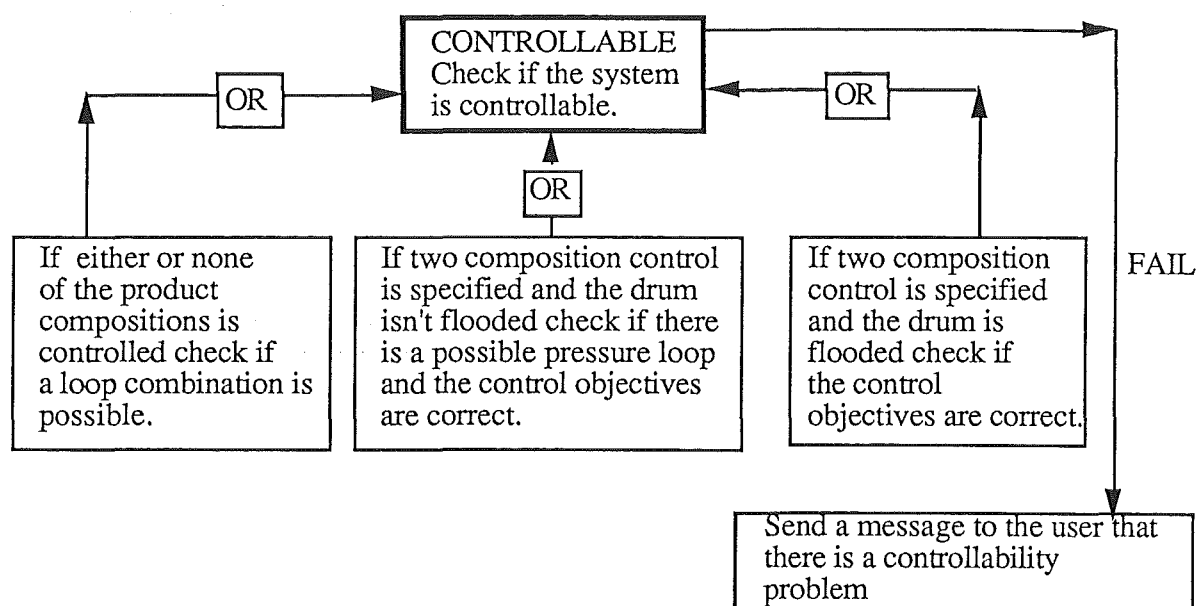


Fig. 19 The structure of the controllability check in the program

3.4 Createloops Section

This part of the program creates all possible combinations that satisfy the controllability criterion from the available pairs and outputs them for the user. It uses list handling predicates for appending variables to and reversing a list. The backtracking capability of Prolog is used to produce all the combinations. The structure of this program subsection is shown in fig. 20.

Some pairings in a control scheme trigger a comment eg. column level controlled by heat input and top temperature by reflux represents vapour to feed control. The program also identifies direct and indirect temperature control schemes and comments on them. This allows the user access to some of the system's knowledge but leaves the final decision on which scheme to use in their hands.

If a column is specified with two composition objectives the program evaluates eight candidate control schemes proposed by Shinskey in terms of relative gain and product size. The structure of this part of the program is the same as the formloops section with success paths created by fired rules if the required facts are available. A success causes the output of the control scheme. The final decision between recommended schemes is left to the user.

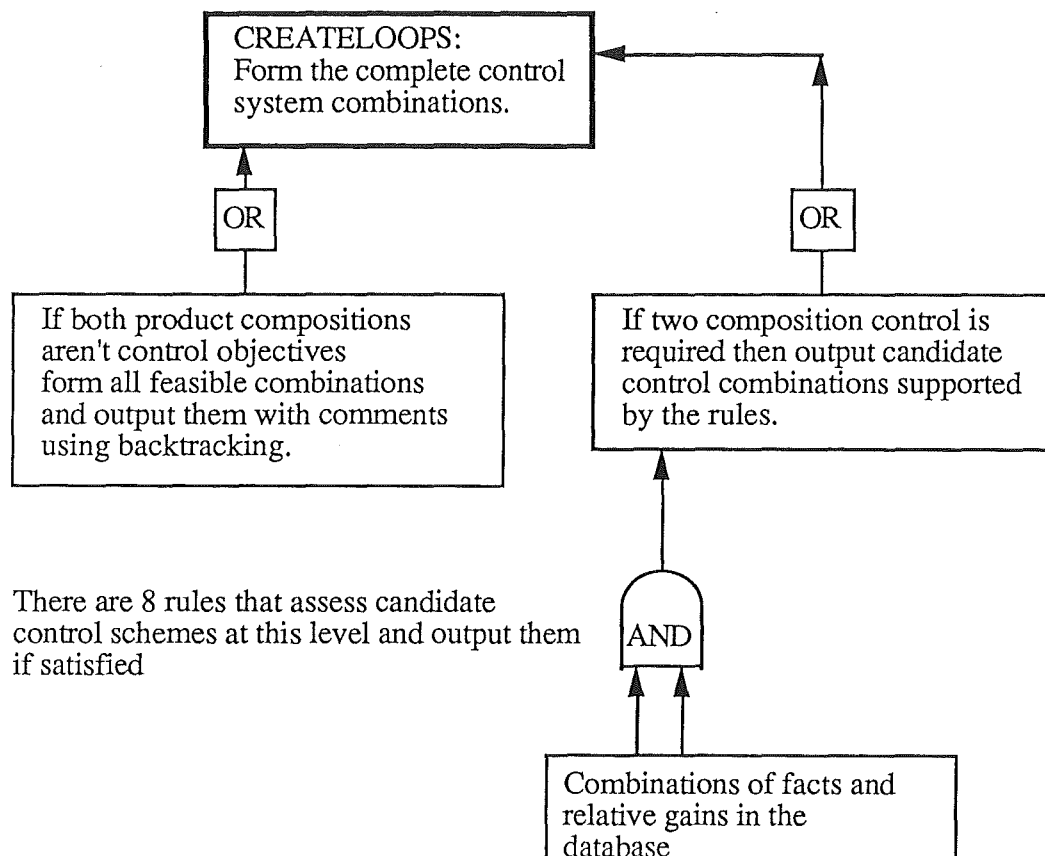


Fig. 20 The operation of the "createloops" section of the program

3.5 The program at work

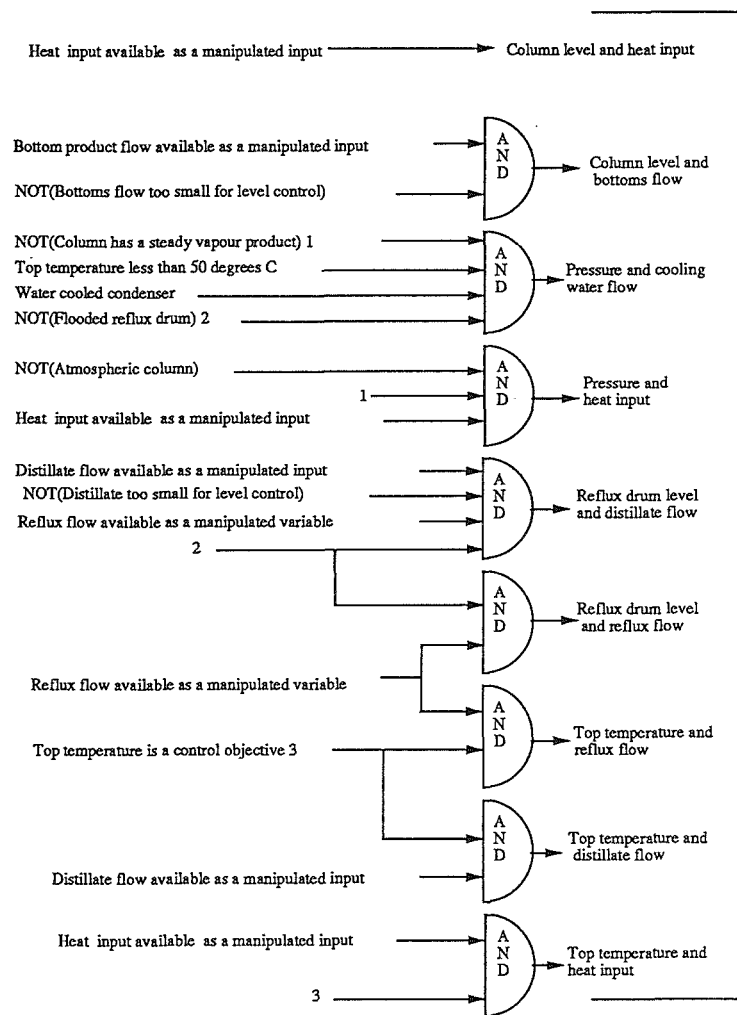
Two examples will be used to demonstrate the program solving a problem. The first example is a single composition controlled column. The feed conditions are unstable (requiring at least one composition control objective), the column has no vapour product, the condenser is water cooled and the column top temperature is less than 50 degrees C. The rules that fire in the formloops section and the output from the createloops section are shown in Fig. 21.

The second example has two parts, both taken from Shinskey (1984). The first part consists of a dual composition controlled column with the fraction of light component in the bottoms, $x=0.01$, the fraction of light component in the distillate, $y=0.99$, the light component in the feed, $z=0.8$, the number of theoretical plates = 50 and reflux ratio = 5. This example has a smaller bottoms flowrate than distillate.

DATABASE SET UP BY THE INPUT SECTION:

POSSIBLE PAIRS:

COMBINATIONS/COMMENTS

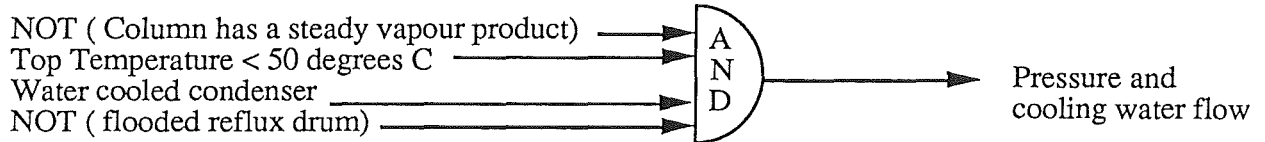


Top Temp	Reflux Drum Level	Column Level	Pressure	Comments
Heat Input	Distillate Rate	Bottoms Rate	Cooling Rate	Direct temperature and pressure control scheme. It offers fastest response to control action and is useful when top product feeds another column.
Heat Input	Reflux Rate	Bottoms Rate	Cooling Rate	Vapour/Feed control which isn't as good as other temperature/pressure schemes.
Reflux Rate	Distillate Rate	Heat Input	Cooling Rate	Vapour/Feed control (see above)
Reflux Rate	Distillate Rate	Bottoms Rate	Cooling Rate	Direct temperature/pressure control scheme (see above)
Reflux Rate	Distillate Rate	Bottoms Rate	Heat Input	Direct temperature/pressure control scheme (see above)
Distillate Rate	Reflux Rate	Heat Input	Cooling Rate	Control scheme has massbalance problems. No product on level control
Distillate Rate	Reflux Rate	Bottoms Rate	Cooling Rate	Indirect temperature/pressure control scheme. It has a slower response than the direct schemes.
Distillate Rate	Reflux Rate	Bottoms Rate	Heat Input	Indirect temperature/pressure control scheme.

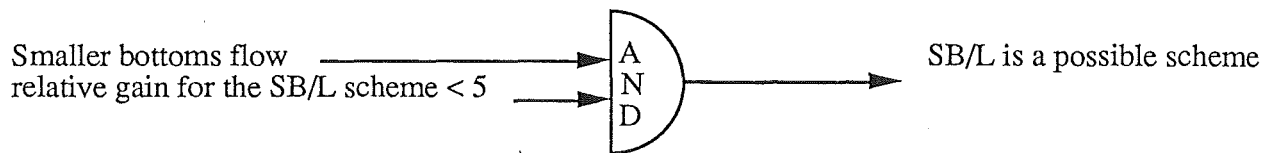
Fig. 21 Program operation on a single composition control example

In the second part $x=0.01$, $y=0.99$, $z=0.2$, number of theoretical plates =50 and reflux ratio =5. This example has a smaller distillate flowrate than bottoms. Both the examples have water cooled condensers and a top temperature less than 50 degrees C. The rules that fire in the formloops and createloops sections are shown in fig. 22.

(i) & (ii) Pressure control alternatives

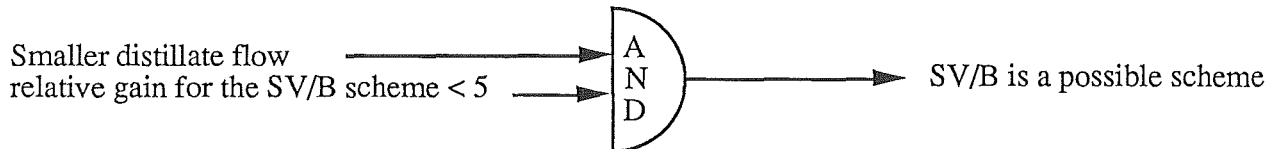


(i) Smaller bottoms flowrate

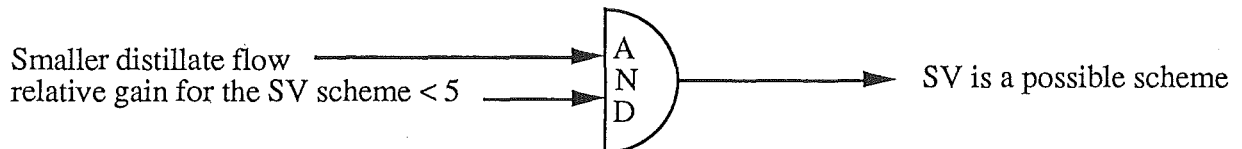


The actual relative gain = 3.29

(ii) Smaller distillate flowrate



The actual relative gain = 1.84



The actual relative gain = 2.09

Fig. 22 The rules fired to solve the two dual composition controlled tower examples

4.0 Nomenclature

B = Bottom product flow

D = Distillate flow

E = Efficiency of an actual separation stage

F = Feed flowrate

L = Reflux flowrate

n = The number of actual separation stages in a distillation column

S = The separation factor

V = Boilup rate in the column

x = Mole fraction of the more volatile component in the bottom product

y = Mole fraction of the more volatile component in the distillate

z = The mole fraction of the more volatile component in the feed

$\bar{\alpha}$ = The mean relative volatility in a binary separation

Λ = Bristol's relative gain between a manipulated variable and control objective

Chapter 3 - Heat Exchanger Control

1.0 Knowledge Acquisition For A Heat Exchanger Expert System

The first section of this chapter is a discussion of the different types of control schemes used industrially for shell and tube heat exchangers. When exchangers are controlled, the objective is usually the target stream exit temperature. However, many exchangers have no control at all and just recover as much heat as possible.

The discussion is in two parts, the first on heat exchangers that don't include a complete phase change in a stream and the second on those that do, because control methods for these two classes differ considerably.

2.0 Heat Exchangers Without an Entire Stream Changing Phase

The control methods described are suitable for exchangers where neither stream changes phase at all or where there is a partial phase change in either or both streams (for example heat exchangers in a methanol production loop where crude methanol is condensed from the gas stream). They are not suited to exchangers where an entire stream condenses (steam heated exchangers).

The temperature control options for exchangers of this type are from two separate families;

- 1) Throttle one of the streams entering the exchanger.
- 2) Partially bypass a stream around the exchanger.

2.1 Throttling Stream Flow

This control scheme is shown in fig. 1 and it is recommended for circumstances where only slow, small range disturbances are expected. There is significant lag and deadtime associated with this type of control and response is

inclined to be sluggish. Shinskey (1979), in his discussion, highlights the non-linearity of this method. There is a continuous variation in the process gain between manipulated flowrate and the heat transferred, as either the manipulated or load flowrate changes. As the manipulated flow increases, the gain decreases and larger and larger changes in process flow are needed to correct upsets to the controlled temperature. This can make valve saturation and loss of control a possibility. Throttling the flow also causes control problems by altering system deadtime. These changing process parameters make tuning a normal PID loop for effective control difficult, unless disturbances to the system are small.

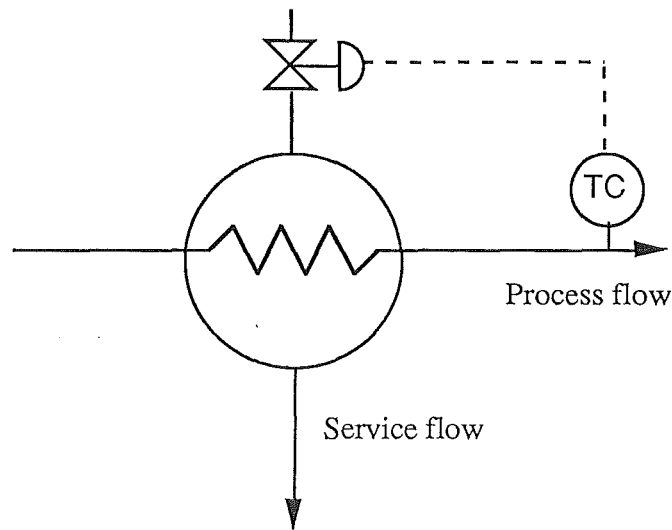


Fig. 1 Stream Throttling Control

The throttled stream shouldn't exceed a temperature that would cause accelerated scaling in the exchanger eg. cooling water temperature must be below 50 degrees centigrade when throttled. If the process stream is throttled then this would only be acceptable if downstream unit operations were unaffected by changes in flow.

The quality of control achievable can be improved using cascade (temperature cascaded onto flow control) or feed-forward control (shown in fig. 2). If disturbances to flow and temperature of the controlled stream are detected before entering the exchanger they can be compensated for earlier than with the feedback scheme. The model is usually a steady-state energy balance around the exchanger,

$$W_1 C_{p1} (T_2 - T_1) = W_2 C_{p2} (T_4 - T_3)$$

and is used to calculate the manipulated flowrate from the measured flowrates, temperatures and setpoint. The control is usually only completely effective when used in conjunction with feedback control of the temperature because the model is never an entirely accurate picture of the heat exchanger. It ignores any heat losses, for example,

and this can cause an offset from the setpoint. The feedback signal can be introduced as a setpoint to the model (replacing the setpoint signal shown in fig. 2) or as an accumulated signal to the control valve (feedforward and feedback controller outputs added together).

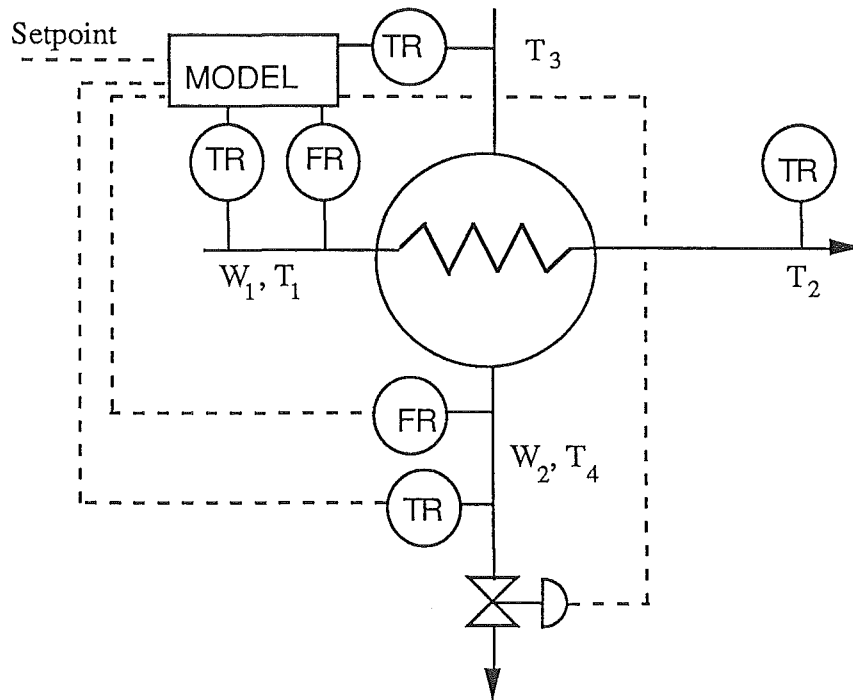


Fig. 2 Feed-forward Control only (without feedback modification)

The feedforward calculation (designated model in the diagram) uses the measured temperatures and flows (W_1, W_2, T_1, T_3, T_4) and the setpoint value for T_2 to find the required service flow signal which is sent to the valve. Perry and Chilton (5th edition) suggest a similar method.

2.2 Bypass Control Methods

The other approach used to control this type of exchanger is to partially bypass flow around the exchanger in response to variations in exit temperature. This approach does nothing to improve the non-linearity of response associated with the heat exchanger, but when the bypassed stream is also the controlled stream the scheme has a faster response than the throttling alternative. The heating or cooling stream can be bypassed but this gives poorer control because of the larger time constant of the system. It may also lead to accelerated exchanger scaling if cooling water is bypassed. The most significant disadvantage that this scheme has is that the exchanger must be oversized in order for it to work (a small but continuous bypass flow reduces the heat

transfer coefficient). In some cases where neither the process flow nor the service flow can be throttled (ie. a downstream reactor or other unit operation requires a steady flow as feed and cooling water would exceed temperature limitations if reduced) there is no alternative but to use this approach.

There are three methods for achieving bypass flow around the exchanger. The usual method is an upstream 3-way control valve. It would be placed downstream only if it was necessary to maintain pressure in the exchanger (to prevent a phase change) and the temperature change of the stream was < 170 degrees centigrade (prevents too large a temperature change across the valve). In cases where temperature is > 260 degrees C or the pressure across the bypass valves is high then 2 x 2-way valves used with a split-range controller is the recommended bypass design (Fig. 3).

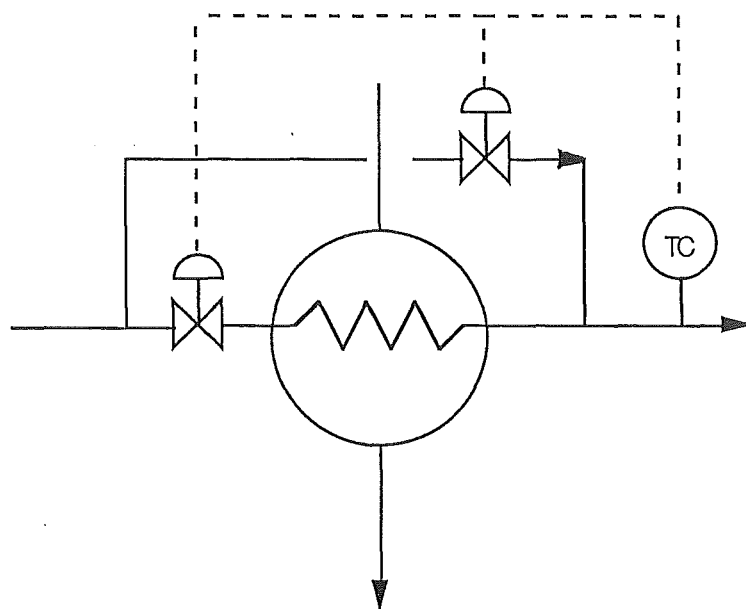


Fig. 3 Two-valve Bypass Control

The simplest and cheapest way of achieving bypass control is a single two way valve in the bypass (Fig. 4) but this requires careful valve sizing. There must be sufficient pressure drop through the exchanger to achieve the flow through the bypass for all expected disturbance conditions or the valve will saturate. The reasoning behind bypass valve selection is explained more fully by Liptak (1985).

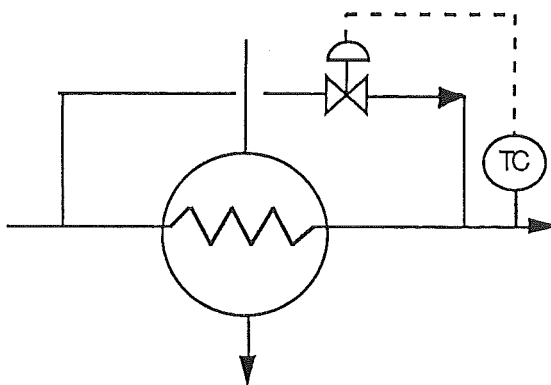


Fig. 4 A Single Two way valve in the Bypass

The control methods for heat exchangers without a phase change are summarised in Table 1 which demonstrates the type of control required depending on the expected disturbance characteristics.

Table 1. Control Methods for Heat Exchangers without a Phase Change

Control Type	Slow disturbances	Fast disturbances
Throttle temperature controlled stream	Reasonable control but this approach is usually not favoured because the process flow must be constant.	Not good control unless the disturbances are in the process flow and can be smoothed using a cascade loop.
Throttle service stream (either heating or cooling)	Reasonable control but has a longer lag than throttling the process flow.	Not good control unless the disturbances are in the service flow and can be smoothed using a cascade loop.
Bypass control	Good control and has a faster response than the stream throttling control alternatives	Reasonable control
Feed forward additions to throttling flow	Good control but a more expensive system.	Good control

3.0 Heat Exchangers With A Phase Change

The most common member of this category is the steam heated exchanger. The different types of control will be discussed under the headings of control type and condensate removal method;

3.1 Control Type

1) Steam valve throttling inlet steam (Fig. 5). This control method works by altering the condensing pressure and therefore the condensing temperature. The tube bundle runs exposed without any condensate covering the tubes. Control quality can be improved by cascade of temperature onto steam flow or heat exchanger shell pressure if these variables are significant sources of upsets (Fig. 6). In order to improve the range of disturbance the scheme can handle a large valve and a small valve can operate in parallel on the steam inlet. The small valve would control if small upsets affected the exchanger and the large valve would open if the change in operating conditions were so large that this was required.

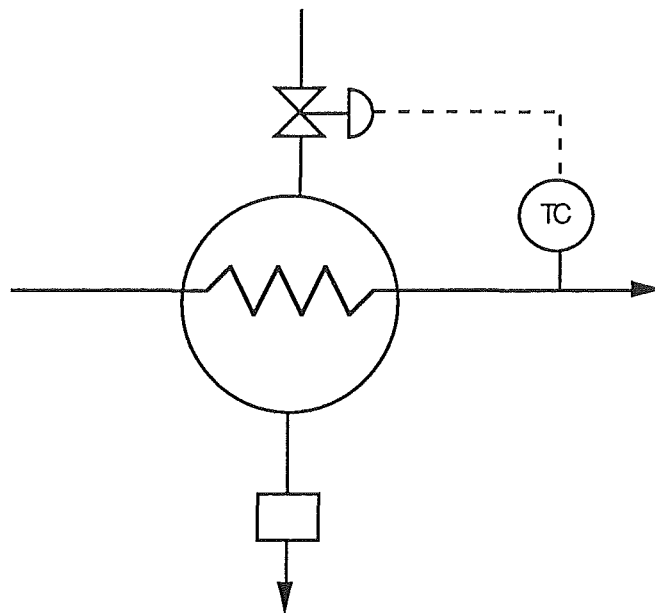


Fig 5. Steam Throttled for Control

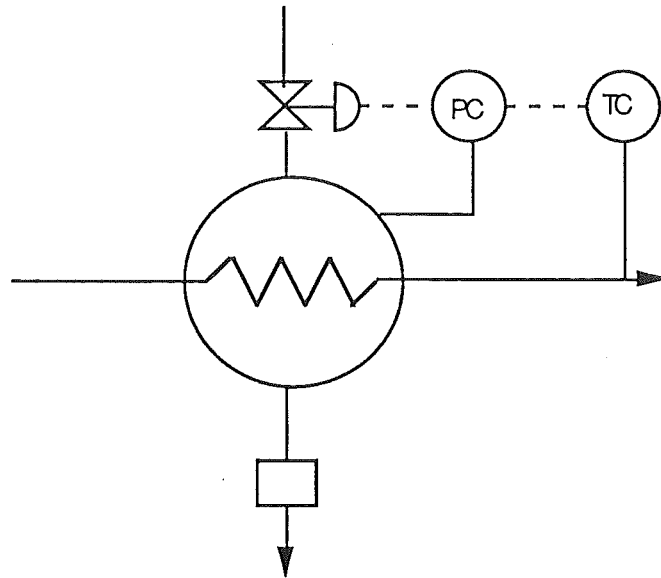


Fig. 6 Steam throttling control with a cascade loop onto shell pressure

2) Temperature control by varying the level of condensate in the exchanger (Fig. 7). This approach changes the amount of exposed area for heat transfer in the exchanger and works because the heat transfer coefficient for condensing is significantly larger than for sub-cooling condensate.

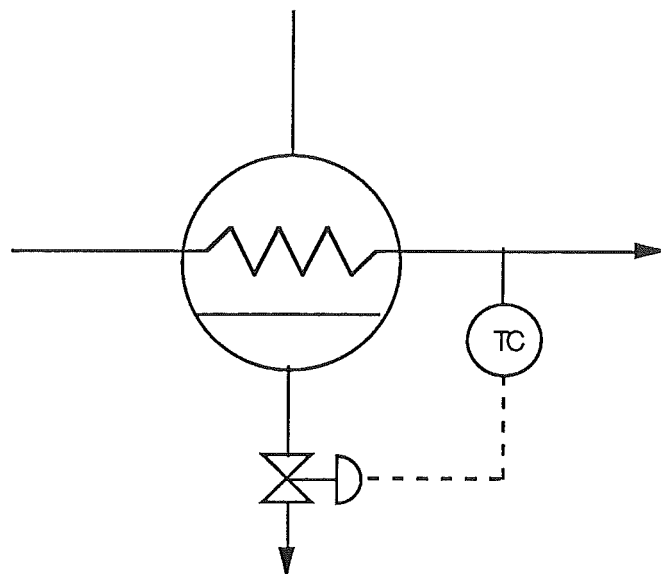


Fig 7. Control by Altering Condensate Level in the Exchanger

3) A mixture of a bypass around the exchanger and steam throttling (Fig. 8). The bypass around the exchanger provides fast response and avoids the heat exchanger dynamics. If a large and sustained change in load variable enters the process, the small range bypass valve may lose control. In order to avoid this the steam valve is manipulated by the valve position controller (VPC) to match the new process

conditions and keep the bypass valve about half open. The valve position controller senses valve top pressure as its measured variable and has a signal corresponding to the bypass valve 50% open as its setpoint. Therefore this scheme provides fast control for a wide range of load disturbance (Allen, 1986 and Perry and Chilton).

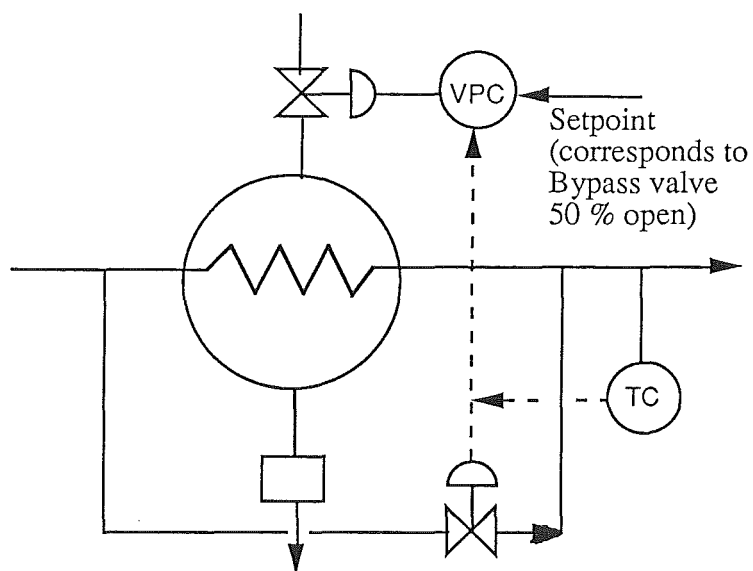


Fig. 8 Bypass/Steam Throttling Combined Scheme (Steam on bypass valve position control)

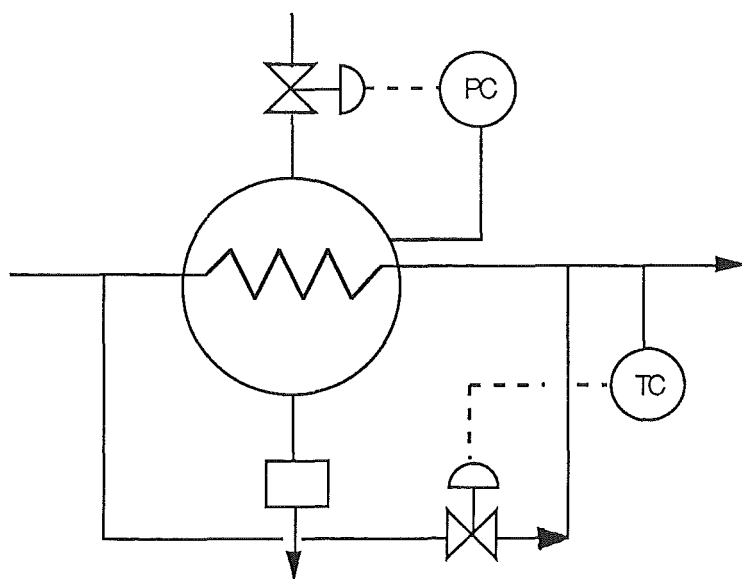


Fig. 9 Bypass control with shell pressure maintained by steam throttling

There is another version (Fig. 9) of the combined scheme that uses steam throttling to control pressure in the heat exchanger. This would be appropriate when it was important to maintain pressure in the exchanger to evacuate the condensate and would achieve a similar control performance to Fig. 8 if operators changed the setpoint of the pressure controller to keep the bypass valve in its operating range (Liptak, 1985).

4) Feedforward from flow and temperature adjusting steam flow to the heat exchanger. This is usually implemented with a feedback trim from the output temperature.

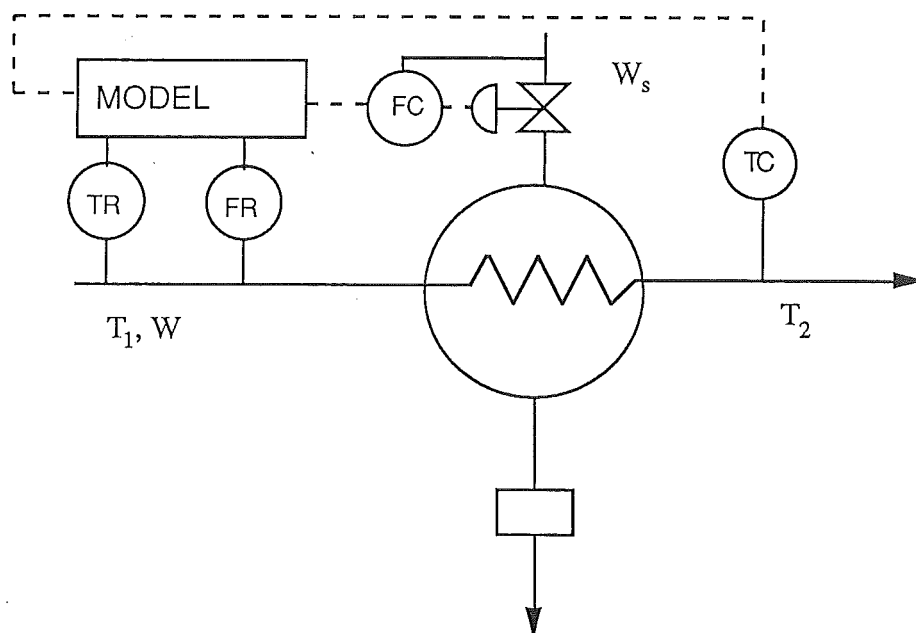


Fig 10 Feedforward/Feedback Control Using Steam Throttling

The model calculates the required steam flowrate from a steady-state energy balance, $W_s H_v = W C_p (T_2 - T_1)$. The model is imperfect so feedback must usually be applied to improve accuracy. In the diagram (Fig. 10) the output from the feedback controller serves as the setpoint signal to the feed forward model (Shinskey, 1979, Liptak, 1985 and Perry and Chilton). Performance is further improved by cascading the signal to the steam valve onto a secondary flow controller. The control methods for heat exchangers with a phase change in one stream are summarised in Table 2.

Control Type.	Small Range Slow Disturbance	Small Range Fast Disturbance	Large Range Slow Disturbance	Large Range Fast Disturbance
Throttle Steam Valve	Should give good control	Control not as good as other techniques. Better with the cascade enhancement	Should consider level control too or valves in parallel on the steam inlet	Control not as good as other techniques. Level control needed. Cascade required.
Level in HX Varied By Condensate Valve	Should give good control. Cheapest method because the valve is smaller than the steam throttling approach	Not good control	Good control on a vertical exchanger	Not good control
Steam Valve & Bypass (Steam valve controlling position of the bypass valve)	Unnecessary expense	Good control	Unnecessary Expense	Appropriate. May require level control.
Steam valve on PC Bypass on TC	Unnecessary expense	Good control	Unnecessary expense	Appropriate. May require level control.
Feedforward with feed-back trim	Improved control	Good control	Unnecessary expense. May need to consider level control	Control good in this case. May require level control.

Table 2. Summary of control methods for heat exchangers with a condensing stream

3.2 Condensate Removal From Controlled Heat Exchangers

There are several methods for condensate removal from steam heat exchangers which are appropriate in different circumstances. If the wrong condensate removal technique is used it may lead to loss of temperature control because the condensate cannot be removed from the exchanger. It is therefore critical to the control scheme.

1) An ordinary inexpensive steam trap (eg. thermodynamic type). This method is appropriate for all types of control except when the scheme requires its replacement with a control valve in the condensate line. Problems can arise when there is a large range of load in an exchanger service or the exchanger is significantly oversized for some operating condition. In these instances the shell operating pressure may run too low to properly evacuate the condensate to the header. Condensate flow slows and liquid builds in the exchanger covering some of the tubes. The heat transfer rate reduces and controlled temperature falls. The steam valve opens forcing condensate out of the exchanger exposing the tubes again and filling the exchanger with steam. The controlled temperature climbs, overcorrecting and causing the steam valve to close, again reducing exchanger pressure and inhibiting condensate removal restarting the cycle. The controlled temperature can cycle out of control and efforts at improving the situation by tuning the controller would be unsuccessful (Mathur, 1973).

2) Pumping trap. This kind of trap is useful for the situation where the condensation pressure is lower than the condensate header pressure (and the condensate must be pumped to remove it from the system). It will operate with the same control methods as a normal trap and it is appropriate for vacuum condensing conditions.

3) Float trap. Works with the same control methods as (1) above but is also appropriate when there are large load changes or the exchanger is oversized for some operating condition. The trap operates because entering condensate raises a float which opens a valve via a lever mechanism and evacuates the trap. It can provide limited level control of condensate in an exchanger.

4) Level Control (Fig. 11). Applies in the same situations as (3) above except this is more widely applicable because it can handle larger load swings due to the ability to achieve greater level variation. It provides a combination solution to a heat exchanger control problem using both level and steam pressure variation which is very

effective but it requires extra investment to include the level controlled vessel (Mathur, 1973).

5) Valve in the condensate line. This is actually a separate control method and is discussed in the control methods section.

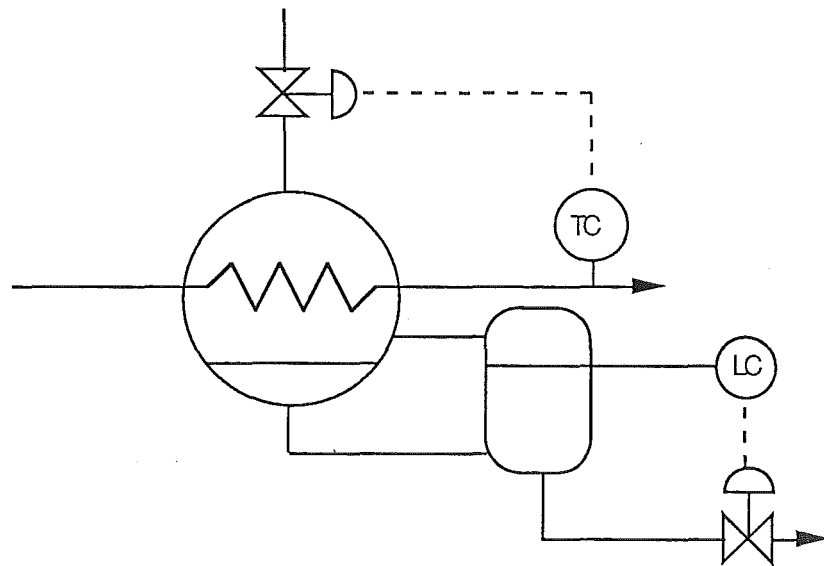


Fig 11 Condensate Removed Under Level Control

4.0 Expert System for Heat Exchanger Control System Synthesis

The program written based around the discussed knowledge handles the two types of heat exchanger;

- i) Shell and tube heat exchangers that do not have a phase change in either stream.
- ii) Shell and tube heat exchangers where the complete condensation of one of the streams occurs (usually steam) to heat the other stream.

The program assumes that the exit temperature of one of the streams is the control objective. This is the usual situation for heat exchangers but represents a different approach from distillation where multiple control objectives must be specified and an important part of the control system design is the selection of the optimum pairings.

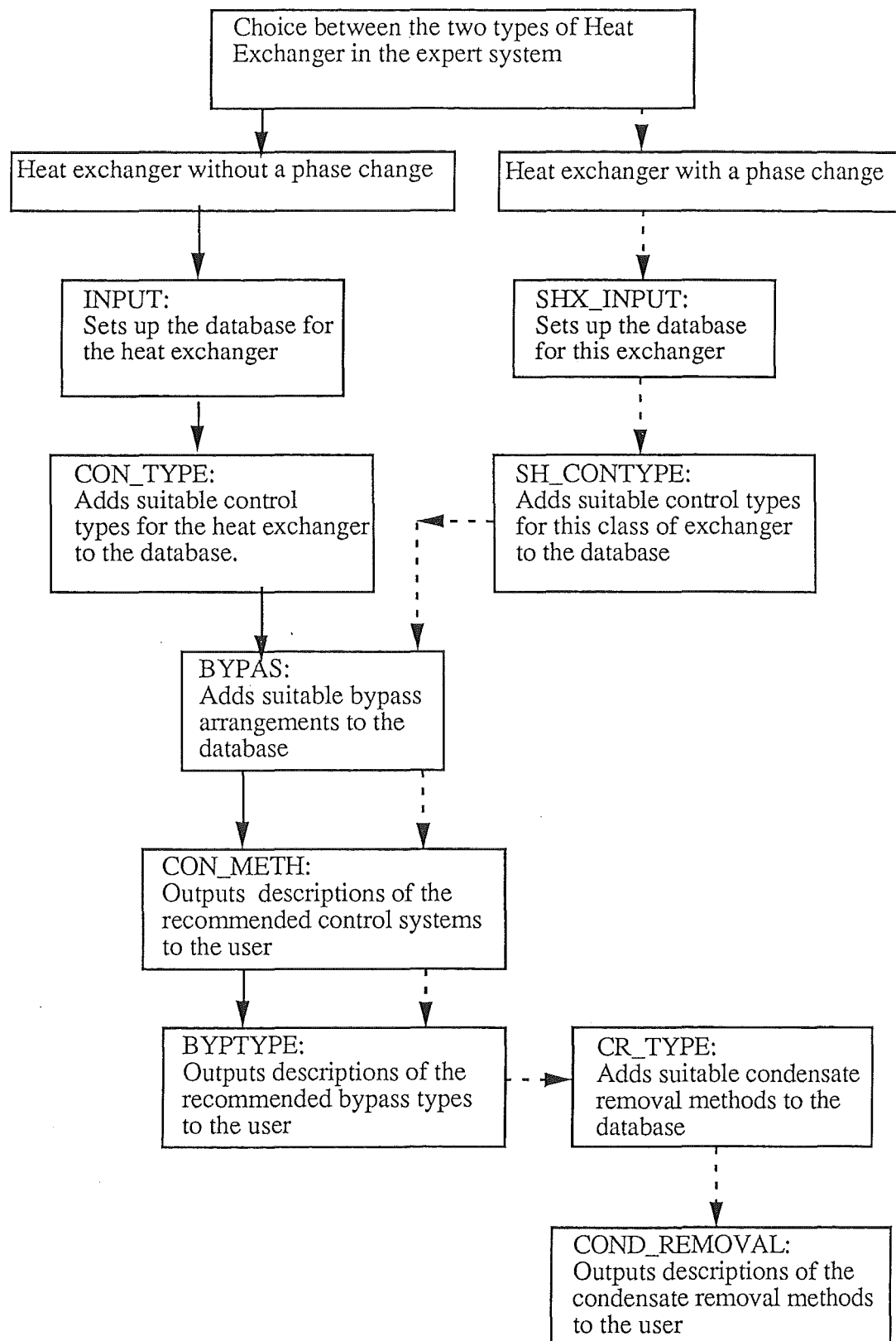


Fig. 12 The structure and sequence of solution of the heat exchanger control systems synthesis expert system.

The program establishes a basic regulatory structure for the exchanger along with some improvements, such as cascade and feed forward control, if the situation demands it. It operates by satisfying a number of subgoals in the order shown in fig. 12.

4.1 Input

This part has a similar structure to the input section of the distillation program (Chpt. 2, Fig. 17). After establishing the basic type of exchanger a series of questions requiring a single word response is followed by a larger set needing yes/no/why replies. The arrangement is the same for both classes of exchanger but the questions and explanations vary depending on type.

As the type of disturbance that affects an exchanger plays a significant role in determining the control method the user must select the appropriate disturbance description from a menu displayed by the program. This choice is important when control methods are recommended later. Tables 1 and 2 earlier in the chapter summarise this relationship between disturbance and control type.

In order to prevent unnecessary questions being asked during the input session each of them has a number of conditions that must be satisfied before they are posed to the user. For example, questions establishing the existence of a bypass around the exchanger are asked before those establishing the availability of streams for throttling. If there is a bypass this is the preferred control method and it is unnecessary to ask whether streams can be throttled for control. The lack of a bypass entry in the database is a precondition to asking whether streams can be throttled and therefore when there is a bypass, the throttling entry cannot be added to the database by the user and the stream throttling solution is never recommended.

4.2 Control Methods

This segment of the program is accessed by the subgoals "con_type" for heat exchangers without a phase change and "sh_con_type" for heat exchangers with a completely condensing stream. The operation of the section is the same as the "Formloops" part of the distillation program. There are a set of rules for establishing control methods for either a heat exchanger without a phase change or one with a phase change. If a rule fires during a scan through the set an identifier for the control method is added to the database. The rules also include a set for bypass design, accessed by the sub goal "bypas", and rules for specifying condensate removal type, under the sub

goal "cr_type". If a bypass valve is incorrectly selected and can't cope with the range of load changes that affect the exchanger or fails because it is in the wrong service (a three way valve in a high temperature process line, for example) control performance is affected. Similarly, it can also be hindered if the wrong type of steam trap is specified to remove condensate (this can cause condensate build-up in the exchanger and a cycling temperature). Therefore rules were included to provide information on the correct type of bypass valve or steam trap required in a particular application.

The rules on control method selection closely follow the tables earlier in the chapter (Tables 1 and 2). For example, a typical rule for a heat exchanger without a phase change is;

" If the disturbances are slow and the controlled stream cannot be throttled and cooling is by cooling water that can be throttled and still remain less than 50 degrees C then control temperature by throttling cooling water flow".

Some of the other rules recommend enhancements to the basic control structure eg. "If the disturbances are fast and expected in the controlled stream flow and the controlled stream can be throttled then control temperature by throttling the controlled stream improving response with a secondary cascade loop". A complete list of the rules is included in Appendix AIV.

4.3 Output Section

The program searches the database for identifiers of particular control methods and, if one is found it matches it with the same identifier tagging a description of the control mechanism which it outputs to the user. If more than one method has been recommended they will all be displayed and the ultimate choice on control method left to the designer. The same process is used to output bypass types and condensate removal methods. The subgoals that begin these actions by the inference engine are "con_meth", "bytype", and "cond_removal" respectively.

4.4 The Program at Work

Two examples will be considered, one from each class of exchanger, to show the operation of the program;

a) An exchanger without a phase change. The following questions and responses make up the interactive input session. The users responses are underlined.

"Is the controlled stream heated or cooled" - cool

"Is the heating/cooling stream a process or utility stream" - ut

expected disturbances identified from the menu of possibilities - fast/small

"Is the major disturbance a flow variation" - y (triggers a menu of stream descriptors)

the disturbed stream is identified as - Cooling water

"Is the utility stream cooling water" - y

"Is there a bypass on the controlled stream" - n

"Is there a bypass on the other stream" - n

"Can the controlled stream be throttled" - n

"Will the cooling water exit temperature be below 50°C if the flow is reduced below design" - y

The input session has identified an exchanger with a cooling water stream that can be throttled upset by fast disturbances with small magnitude. The program suggests two possible solutions through the firing of the rules shown in fig. 13.

i) Feed forward control throttling the cooling water flow

ii) A cascade of the primary temperature control loop onto flow control of the cooling water stream.

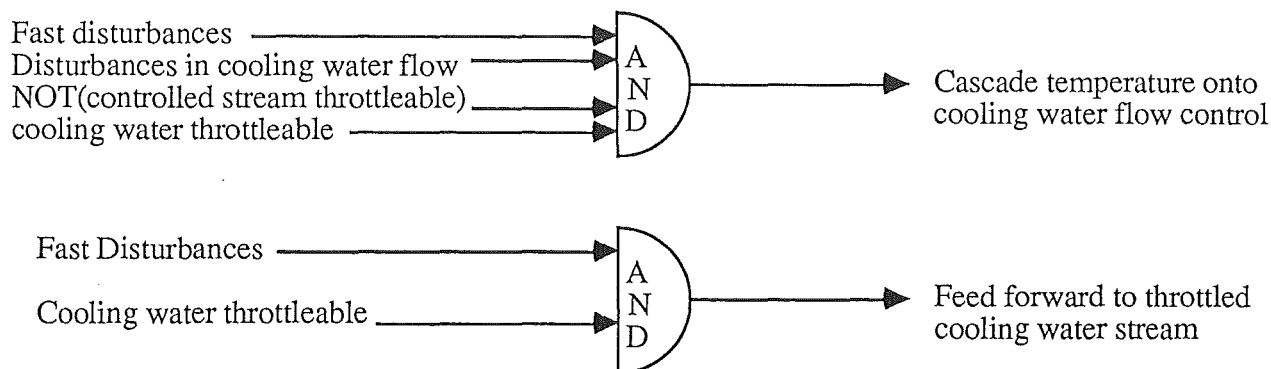


Fig. 13 The rules fired when recommending control for the heat exchanger without a phase change

b) Heat exchanger with a completely condensed stream. The following questions and answers are asked during the input session.

The disturbance type is identified as - Fast/large

"Is there a bypass on the temperature controlled stream" - y

"Is there sufficient driving force for a single bypass valve" - y

"Is the operating temperature $> 260^{\circ}\text{C}$ or valve pressure drop high" - y

"Is the condensing pressure $<$ Condensate removal header pressure" - n

"Is the exchanger oversized for an expected operating condition" - n

The program recommends controlling temperature using the bypass valve holding this valve on control by varying the condensing stream (steam) valve position. It recommends a single two way valve in the bypass and a drainer trap or a level controlled vessel to remove condensate. The rules fired in the inference are shown in fig. 14.

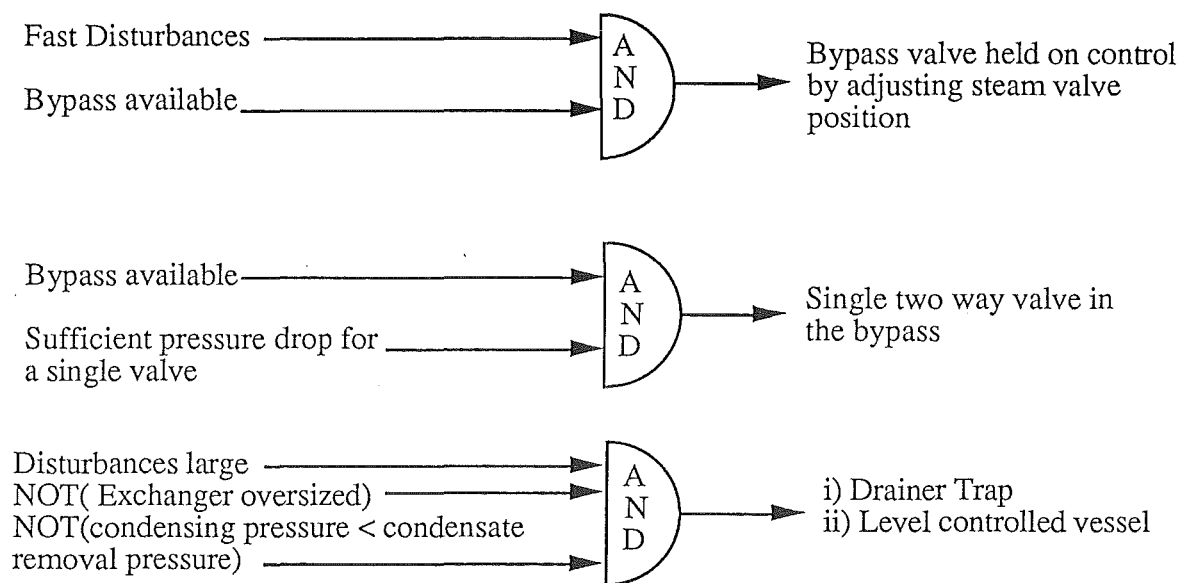


Fig. 14 Rules fired in analysing the example of a heat exchanger with a condensing stream

5.0 Nomenclature

W = Flowrate of a heat exchanger stream

C_p = Heat exchanger stream specific heat

T = Stream temperature

Chapter 4 - Reactor Control

1.0 Knowledge Acquisition for a Reactor Control Expert System

The expert system developed recommends control schemes for reactor systems commonly used in industry, the Continuous Stirred Tank Reactor (CSTR), Tubular and fixed bed catalytic reactor. Before considering each of these pieces of equipment in isolation a discussion of the the consequences of fundamental physical reaction data on control is included.

As most reactions are energy sensitive temperature control is the most important aspect of a reactor control scheme. A large part of the discussion in this chapter, and the expert system itself, is consequently devoted to temperature control.

1.1 Fundamental Reaction Data and Control

The reaction type plays a critical part in deciding the equipment used for the reaction and therefore the control scheme required.

1) Endothermic reactions require a heat source of some kind to supply the heat of reaction and the equipment reflects this. Methane or natural gas reforming is normally carried out in a furnace operating at the constraints of tube metal and refractory maximum temperature. The control problem is to control the firing in the furnace to achieve the required reaction rate and to monitor and control the excess oxygen and carbon monoxide levels in the flue gas produced during furnace firing. A cracking furnace in an ethylene plant has a similar requirement although in this case the selectivity of the reactions occurring is important (often called cracking severity). If steam is generated, boiler feedwater flow and drum level control are also important. The schemes used in this example are very similar to those used for boiler control. Endothermic reactions are self regulating in temperature, unlike exothermic reactions.

2) Exothermic reactions require a cooling source under normal operating conditions and yet also require an initial source of heating to reach reaction temperature in many cases and thus temperature control becomes very important. The question of

reactor stability has to be addressed carefully during the design of exothermic reactors, especially for irreversible exothermic reactions (see fig 1.). Reversible reactions slow as they approach equilibrium and don't 'runaway' but they may exceed equipment temperature limits if not controlled carefully.

Points A and C in Fig. 1 represent stable steady states whereas B is unstable at the second of the two heat removal rates because a small increase in temperature increases heat evolution to a greater extent than the heat removal system can handle.

Operation at B can be stabilised by redesigning the system to run at a higher jacket temperature, by increasing heat removal capacity (this is demonstrated by the first heat removal curve in the diagram) or by feedback control. Exothermic reactions are also extinguishable ie below a certain ignition temperature they proceed very slowly.

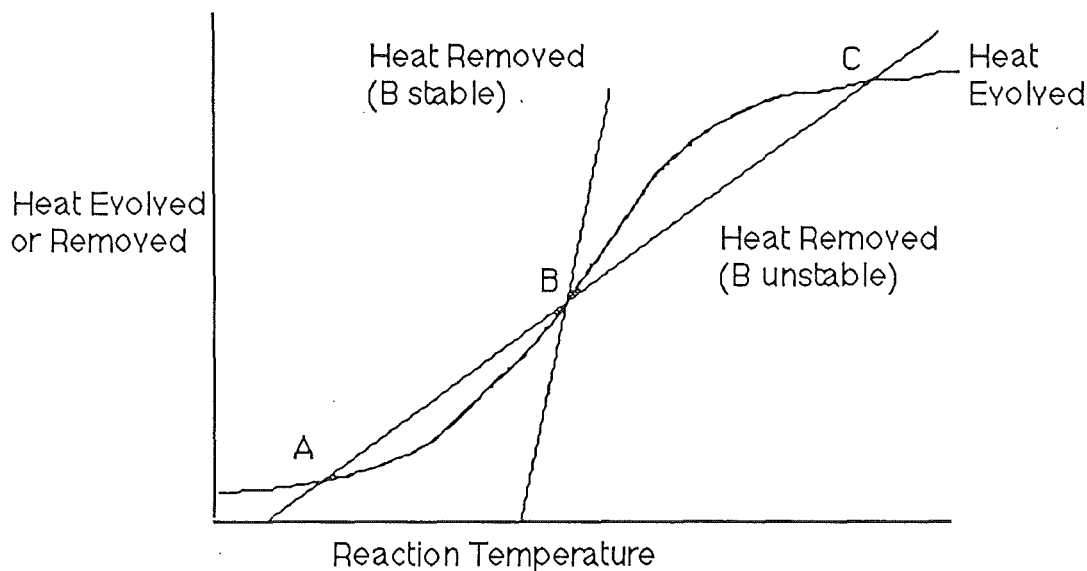


Fig. 1 Heat evolved in an exothermic reaction and heat removed at two different cooling rates

3) The phases involved in the reaction influence equipment and control scheme. Liquid phase reactions can be carried out in all three types of equipment whereas gas phase reactions occur in tubular or fixed bed reactors. Mixed phase reactors have differing control schemes to single phase reactors. Gas/liquid reactions for example will often have the gaseous reactant added on pressure control and the liquid product removed under level control (see the "mixed phase reactions" in the CSTR section).

4) Reactor kinetics can be used to give information as to which concentration should be controlled to affect reaction rate. This is important in the discussion of the control scheme for the Oxo reactor.

5) Whether a reaction is reversible or irreversible plays a significant part in deciding the form of the equipment used. For reversible reactions the composition at the exit of the reactor can be governed by equilibrium conditions whereas irreversible reactions can proceed to completion (dependent on rate and inlet concentrations of reactants). Reversible reactions often use the reactor-product separation-recycle-fresh feed addition form of plant that occurs so often for reacting systems.

6) Most industrial reactors have more than one reaction occurring at any time and for this situation selectivity becomes important to ensure the correct product slate. This can often be met by controlling reaction conditions such as pressure and temperature.

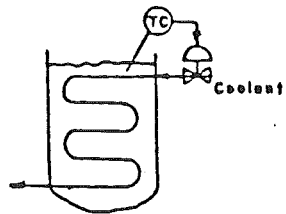
7) Catalytic reactions often have constraints that apply because of catalyst limitations or poisoning. There must be, for example, a certain amount of hydrogen in the top of a reforming furnace to prevent methane cracking and coking the catalyst. The steam to carbon ratio must also be kept high for the same reason. Catalyst will age at differing rates depending on temperature and this will often set an operating range for a reactor. Temperature must generally be controlled within this range for a catalytic system.

In summary, the basic chemistry of the reacting system decides what type of reactor will be used and this has a strong influence on the control scheme required. These conditions often also decide what the control objectives are for the control scheme.

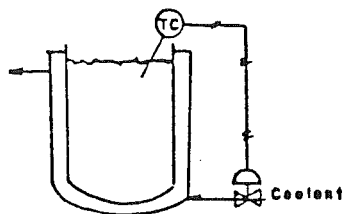
2.0 Continuous Stirred Tank Reactor Control

2.1 Temperature Control

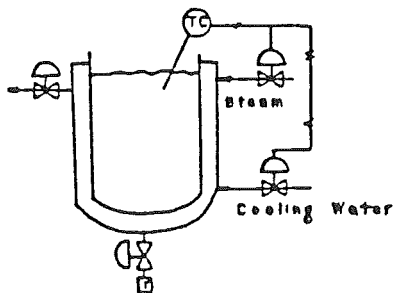
Much of this discussion and the types of reactor are taken from the reference by Luyben (1981) that discusses a number of different configurations for cooling exothermic reactions in CSTR's and the typical control systems for each. The reactor types are shown in fig. 2.



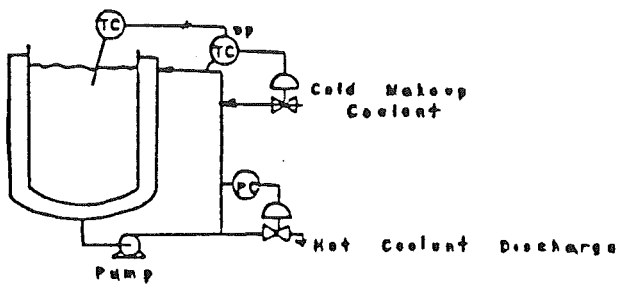
A. Cooling Coil



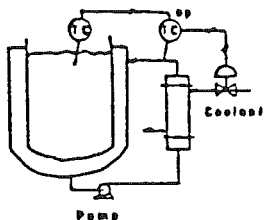
B. Cooling Jacket



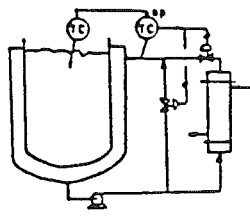
C. Jacket With Cooling or Heating



D. Circulating Water System With Direct Addition of Coolant and With Jacket Temperature Cascade Control

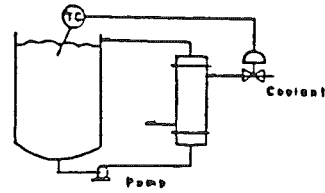


E1

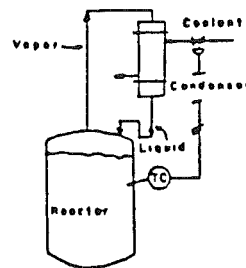


E2

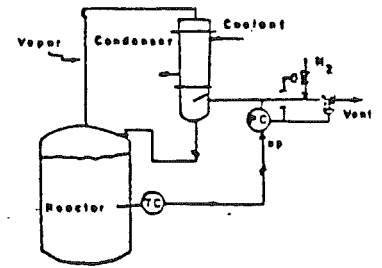
E. Circulating Coolant With External Cooler



F. Circulation of Reaction Liquid Through External Cooler

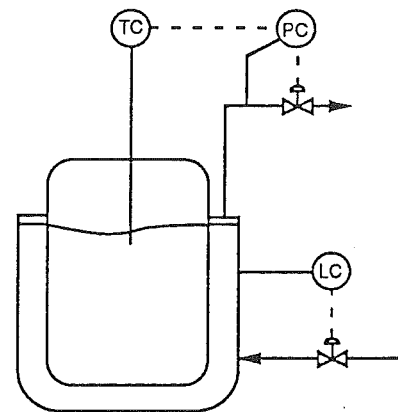


H1



H2

H. Boiling liquid reactors



G. Steam production in the jacket

Fig. 2 Temperature control methods for continuous stirred tank reactors.

1) When the heat evolution rate is small reactors using either a cooling coil (Fig. 2, A) or jacket (Fig 2, B) with once through coolant flow manipulated for temperature control are satisfactory. If the reaction is more highly exothermic there is a difficulty in obtaining sufficient heat transfer to ensure stability.

2) A jacketed CSTR with either steam or cooling water in the jacket. The typical control scheme is a split range temperature controller manipulating either steam or water flow to the jacket (Fig. 2, C). This configuration is used when the reaction mixture must initially be heated and is more suited to batch than continuous reactors. In the cooling phase the control characteristics are the same as (1) and the same comments apply.

3) A jacketed CSTR with coolant circulated at a higher rate than in B using an external pump (Fig. 2, D). A cascade of the primary temperature control loop onto the temperature of the coolant entering the jacket improves the system response to disturbances in coolant temperature (controlled by a make-up stream of fresh coolant). Hot coolant is discharged from the circulation loop under pressure control This system is capable of higher rates of cooling than a once through jacket, because of the higher heat transfer coefficient and increased temperature difference that are a consequence of more coolant flow.

4) A jacketed CSTR with an externally pumped coolant flow and an external heat exchanger used to cool the circulating flow. The reaction temperature is controlled by throttling the cooling stream in the heat exchanger (Fig. 2, E1). An improvement to provide better response to cooling water inlet temperature variations is to cascade the primary temperature control loop onto a secondary loop controlling inlet temperature. This approach has similar advantages to (3) and is used when the circulating coolant must be some type of heat transfer fluid, because the reaction temperature is significantly higher than normal cooling water temperature. This helps to ensure reaction stability and also provides a very effective way of preheating the reaction mass.

5) Essentially the same configuration as in (4) but with a bypass around the exchanger with two valves manipulated by an overlapping range temperature controller (Fig. 2, E2). This system removes the inevitable lag associated with manipulating cooling flow to the exchanger and provides rapid control action.

6) The reaction mixture is circulated through an external heat exchanger for cooling. The temperature controller throttles the coolant flow to the exchanger (Fig. 2, F). This allows higher heat transfer area availability than a jacketed CSTR but isn't

recommended when corrosive materials, slurries or polymers make up the reaction mixture.

7) A jacketed reactor generating steam in the jacket. The feedwater is added on jacket level control and the temperature loop is cascaded onto a jacket pressure controller manipulating a valve in the exit steam line. A high pressure steam source directly into the jacket is an addition that can provide reaction mass preheat. This method has a very fast temperature response and very high heat transfer coefficients and therefore can cope with more exothermic reactions (Fig. 2, G)

8) A boiling liquid reactor where the reaction is maintained at constant temperature as long as the system operates at constant pressure. The heat of reaction vaporises the reaction mixture or a solvent added because its boiling point is a reasonable reaction temperature. The vapour is condensed and refluxed back to the reaction mixture. The simpler control scheme throttles coolant flow to the condenser which corrects changes in temperature by altering the reactor pressure (Fig. 2, H1). The boiling liquid reactor is suitable for highly exothermic reactions because the heat transfer surface is effectively right through the reaction mass and the reactor is self regulating and stable.

9) The same equipment as (8) but here the temperature controller is cascaded onto an inner pressure control loop. The pressure controller manipulates two valves, one in the vapour product line and the second on an inerts feed to the condenser, using split range control. This method relies on there being both a vapour product and a source of inerts (Fig. 2, H2).

If the reaction is endothermic temperature is far less important as a control objective. Reactant conversion is usually constrained by the amount of heat that can be supplied to the reactor and its temperature is stable and not prone to runaway as an exothermic reaction's can.

2.2 CSTR Stability

An unstable operating point for a jacket cooled CSTR occurs when the slope of the heat production curve is greater than the slope of the heat removal curve at their intersection. This situation happens when;

$$RT^2/E > T - T_j \quad (\text{Harriott, 1964})$$

where R=gas constant; T=reaction temperature; E=activation energy; T_j =jacket temperature. Operation at an unstable point is possible if the temperature is controlled

by feedback using flow to the jacket or jacket temperature and the proportional gain falls between a minimum and maximum value. This approach won't work if the temperature measurement and heat transfer time constants are significant when compared with the process time constant (Luyben, 1981).

2.3 CSTR Composition Control

In any chemical reactor, including the CSTR, if the reaction conditions remain constant and there are no disturbances in feed rate and composition then the product will also have a constant composition. When this is not the case however some form of composition control may be necessary. This may take the form of manipulating reactant feedrate (this indirectly affects rate by changing the concentration of reactants in the reactor), manipulating the holdup in the reactor (changing level perhaps), altering catalyst concentration or temperature. Experience from the literature (Harriott, 1964) indicates that this appears to be difficult to automate. The alternative is to accept what you get from the reactor in the face of changes. This is often a viable solution, the reformer and converter in a methanol plant work this way and the only concession to composition control is manual adjustment of reaction conditions by operators.

2.4 Inventory and Flow Control for a CSTR

If the reaction temperature is under control the feed flows, product flow and inventory must also be constant around the reactor to maintain conversion at the required level. Shinskey (1979) includes a discussion about these control considerations for several different classes of reactor.

2.4.1 Mixed Phase Reactions

If gaseous and liquid reactants are mixed together to produce a single gaseous product in a CSTR then the gaseous reactant can be added on flow control, the liquid added on level control and the gaseous product removed under pressure control (a small manual liquid purge is required to remove any inerts that accumulate in the vessel liquid).

Alternatively if a liquid is reacted with a gas to produce a liquid product the liquid can be added under flow control, the gas added under pressure control and the liquid product removed under level control (a small manual purge of inerts from the gaseous head space is required to prevent their accumulation).

2.4.2 Single Pass Reactors

If there is more than one feed to a single pass reactor (usually two) there should be careful control over the ratio of the reactant feed rates. Both the feed streams should be on flow control, with a ratio controller supplying the set point to one of the flows. If the composition of either feed is variable, the ratio between the two streams should be trimmed by an exit composition analyser. Another possibility that may exist is that one of the feeds enters the plant from an upstream operation and there can be no flow control on it. In this case, the flow should be measured and the other controllable flow altered in ratio to it. As this may result in varying flow through the reactor the control scheme should have an extra calculation block that senses total feed flowrate and calculates the setpoint for the level controller on the product stream. Reactant conversion should be maintained by this approach. The risk is that the reactor may overflow if the feed flowrates increase, so to complete the control system there should be an override on the setpoint signal sent to the level controller. This is discussed in the example of the modified Williams-Otto plant in Chapter 6.

2.4.3 Recycle Reactors

There are two different types of recycle reactor. In the first (Fig. 3a), if a reactant is recycled it is always in excess and control of its flow isn't critical as reaction rate is determined by the concentration and therefore feed flow of the other reactant. The required make-up of the excess reactant is added to the recycle storage tank under level control.

In the second (Fig. 3b), if a solvent is added or product recycled to moderate a reaction, then a similar style of control can be used as applied to the single pass situation. The solvent is separated from the product mixture and recycled to the reactor. If some of the reactant remains dissolved in the solvent and is returned to the reactor the ratio between the two feeds must be trimmed using a composition controller sensing the concentration of the excess reactant in the product stream.

3.0 Plug Flow Reactors

This class of reactors includes tubular reactors or collections of them in parallel (a steam reformer for example) either empty or filled with catalyst, jacketed for exothermic reactions. It also includes heat exchanger type reactors normally with catalyst packed in the tubes and a coolant circulated through the shell. The control considerations are the same as for the CSTR.

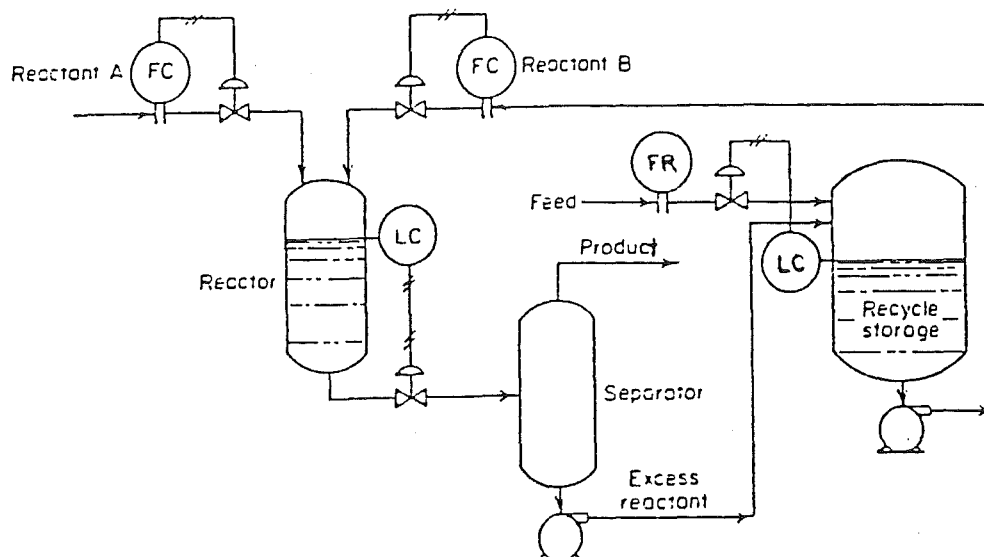


Fig. 3a The flow of reactant B may be set to limit the concentration of reactant A or to fix the residence time.

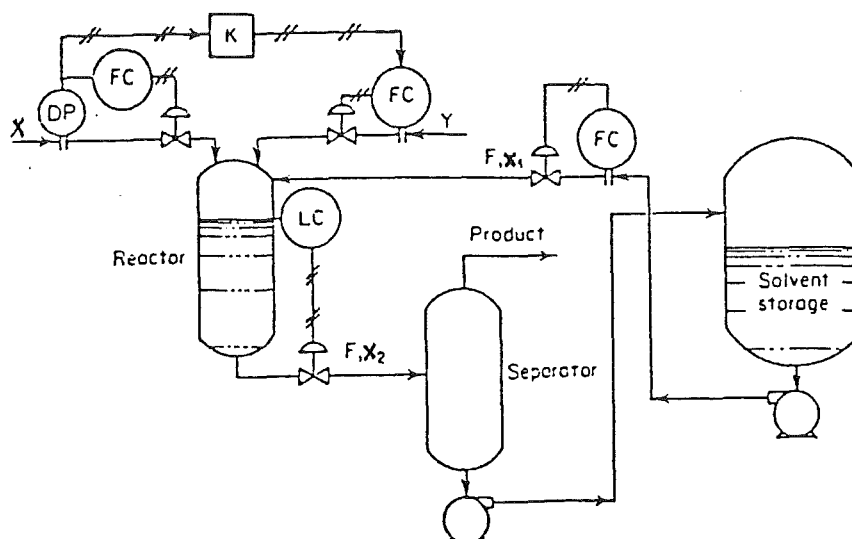


Fig. 3b Excess reactant is ordinarily recycled with the solvent

Fig. 3 Continuous Stirred Tank Reactors in a recycle system
(Shinskey, 1979)

3.1 Temperature Control

Once again temperature control is critical for exothermic reactions. It is also more difficult than in a CSTR, which has a significant thermal inertia, because there can be rapid changes in temperature profile when small disturbances upset the process especially in tubular reactors that are not packed with catalyst.

The typical control configuration for a jacketed tubular reactor is to have a primary reaction temperature controller cascaded onto a secondary jacket temperature controller manipulating the flow of coolant through the jacket. This approach has similar limitations to the version used for cooling a CSTR and is really only suitable for reactions which have a small heat release. If the reactor is a heat exchanger type and the reaction is carried out at a high enough temperature in a plant with a compatible steam system steam can be generated in the jacket to remove the heat of reaction. The reaction temperature is controlled using a primary controller cascaded onto a secondary jacket pressure controller (this is shown in the diagram of the Oxo reactor in Fig. 4). This configuration has higher heat transfer coefficients than the jacketed version and is suitable for more highly exothermic reactions

The other class of continuous reactions are endothermic and the reaction vessel is often a fired heater of some kind (eg. reformer or a pyrolysis furnace in an ethylene plant). In this case the reaction temperature is often monitored rather than controlled and the actual reactor control scheme centres around reactant ratio control eg. steam to carbon ratio in a reforming furnace. The other control considerations relate to furnace control (see the section on endothermic reaction control under the discussion on fundamental considerations). This generally decides the extent of reaction or the reaction product mix (for the ethylene pyrolysis furnace).

3.2 Stability of Plug Flow Reactors

The temperature profile in a plug flow reactor rises to a maximum value a short distance into the reactor. There is a similar stability criterion for tubular/catalytic reactors as CSTR's. If the difference between reaction and jacket temperature is close to the critical temperature difference ie. $T - T_j = RT^2/E$ then the maximum value changes significantly for very small changes in inlet temperature or reactant concentration. A new steady state profile exists for this condition so a reaction runaway doesn't occur. However it is impractical to design a control system to maintain reaction temperature below constraints imposed by reactor material specifications and catalyst operating limits. Therefore a conservative design criterion for tubular reactors is that $T - T_j <$

RT^2/E . The actual limits can slightly exceed this in some circumstances. Harriott (1964) has a detailed derivation of these values.

Another form of instability occurring in reactors of this type happens when the product from an adiabatic reactor preheats the feed. If a small increase in feed temperature occurs then the exit temperature increases which in turn increases inlet temperature. This positive feedback effect could lead to a reaction runaway.

3.3 Composition Control in Plug Flow Reactors

In order to control conversion in a plug flow reactor the temperature profile and feed conditions must be steady. In the case where there is a single feed to the reactor it should be flow controlled to prevent variations in feed upsetting the reaction and if there are two feeds then they should be flow controlled with ratio control between the feeds. An example of this approach is shown in Fig. 4 where the synthesis gas and the propylene are fed at a controlled ratio of 3:1 to the Oxo reactor. The steam and methane feeds to a reforming furnace are also fed in under ratio control. In instances where there is a catalyst feed available it can be varied in response to variations in feedrate or composition to alter the rate of reaction and maintain the exit composition from the reactor constant.

Plug flow reactors are often used for gaseous reactions and may form part of a recycle loop. The operating pressure of such a loop has an affect on the reaction rate as does the composition of the recycled stream. Pressure in the loop is often controlled by varying a small purge flow from it. The rate of purge flow is governed by the concentration of inerts entering the loop as these must be removed at the same rate as they are fed or they will build-up in concentration and affect the rate of reaction as well as causing an increase in pressure.

4.0 Fixed Bed Reactors

The fixed bed reactor is the final class of reactor considered. These are commonly continuous flow, multi catalyst bed devices (adiabatic reactors) with provision for heating or cooling the reacting mixture between the beds. This form of reactor is normally used in gaseous reactions.

4.1 Temperature Control of Fixed Bed Reactors

If the reaction is exothermic the type of reactor decides the control scheme. For example, if there is a low temperature feed before preheating then interbed cooling is economically and directly applied by injecting some of the low temperature gas at the exit of the catalyst beds. This is known as a " quench reactor " and common industrial examples are the ICI Methanol reactor and the Ammonia Quench reactor.

In instances where the feed can't be used to quench the reaction a more expensive alternative is to pass the exit stream from a catalyst bed through an external heat exchanger to cool it before it enters the next bed in the reactor. If the reaction is endothermic the same style of equipment is used but the interbed heat exchangers supply rather than remove heat from the reacting mixture.

4.2 Composition Control in Fixed Bed Reactors

Fixed bed reactors are often used in the same services as tubular reactors ie. gaseous reactions in a recycle loop and the same control considerations apply. Composition is controlled by maintaining the temperature profile in the reactor and keeping the feeds on flow control and in the correct ratio. Pressure control is often achieved by manipulating a purge bleed to control inerts concentration in the loop.

5.0 Industrial Case Study of The ICI Oxo Reactor

This case study was taken from the report written on modern control system applications in Australia produced by the Warren Centre at Sydney University (Weiss et al, 1987) and demonstrates the approach an expert system should be able to take when faced with the problem of designing a control system for a reactor.

The Oxo reactor is a plug flow reactor with a free volume of 1.8 m^3 fitted with cooling tubes connected in closed circuit with an overhead steam drum to remove the exothermic heat of reaction. Propylene liquid, synthesis gas and cobalt catalyst slurry are fed into the bottom of the reactor. The control objectives are to maximise the total butyraldehyde production rate and/or the n:iso butyraldehyde ratio in the raw oxo product.

The major disturbances that enter the reactor are upsets in catalyst slurry concentration and feed rate variations in flow and composition. The control scheme originally involved reactor pressure controlled by syngas flow, reactor coolant

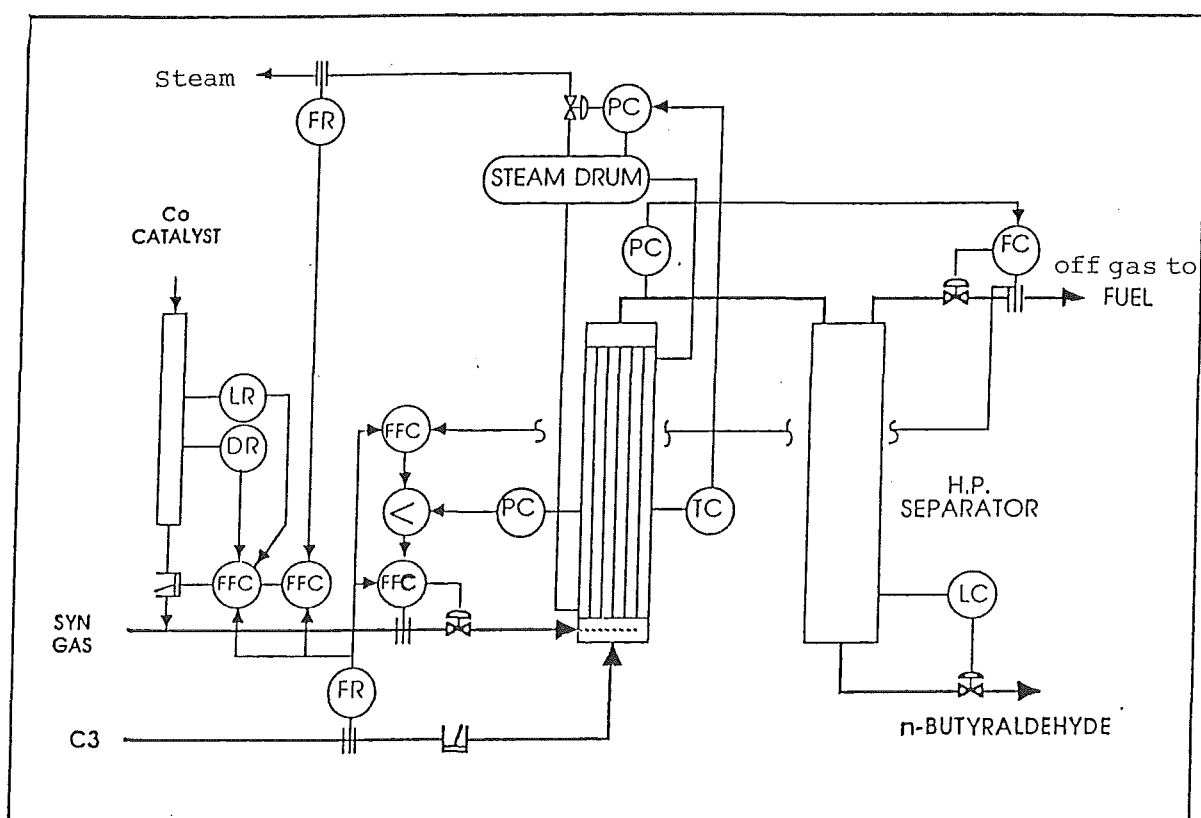


Fig. 4 Oxo Reactor control scheme (Weiss et al, 1987)

pressure by vent valve position, offgas on flow control and product on level control from the separator. There are rapid fluctuations in syngas rate because it is fed under pressure control. It is known that higher reaction temperatures lower the n:iso ratio so that the reaction temperature must be kept as low as possible without extinguishing the reaction. The propylene rate is kept as high as possible to maximise production rate. Syngas feed is fed in at an appropriate ratio to the propylene (3:1). Offgas purge needs to be kept as low as possible to avoid syngas waste but still remove the inerts fed to the reactor. Low temperature can cause Cobalt deposition and reactor instability. The original scheme proved inadequate and the revised control scheme was as detailed below (fig. 4);

i) The reactor pressure controlled by a cascade onto an inner flow control loop on the offgas rate. If valve saturation occurs then the syngas to propylene flow ratio is adjusted to reduce pressure and bring the offgas flow valve back into its control range

ii) The offgas to propylene feed ratio is controlled by measuring the offgas flowrate and adjusting the syngas/propylene ratio controller setpoint using a slow loop.

iii) The reaction rate is inferred by the steam production rate. The cobalt catalyst addition rate is manipulated to control steam production rate and therefore reaction rate. Disturbances in cobalt concentration are measured using a density correlation and compensated for by altering the catalyst addition rate.

iv) The reaction temperature is also controlled because its variation affects reaction rate. A cascade loop of temperature onto a pressure controller manipulating a valve in the steam vent line is used.

The revised control system was shown to have a superior performance to the original in a series of dynamic simulations. All the control loops were implemented as PI controllers, a very standard industrial implementation (improved control can be attained without using modern control techniques in this case). The overrides shown in the diagram are at this point beyond the capability of the expert system.

6.0 The Reactor Control System Synthesis Expert System

The reactor expert system, developed from the discussion on reactor control, was written in three separate modules;

1) REACT1 - selects temperature control methods for continuous stirred tank reactors.

2) CSTR - selects inventory and composition control methods for continuous stirred tank reactors.

3) TUBFBR - selects temperature, composition and inventory control methods for tubular and fixed bed reactors. The overall program structure is shown in Fig. 5.

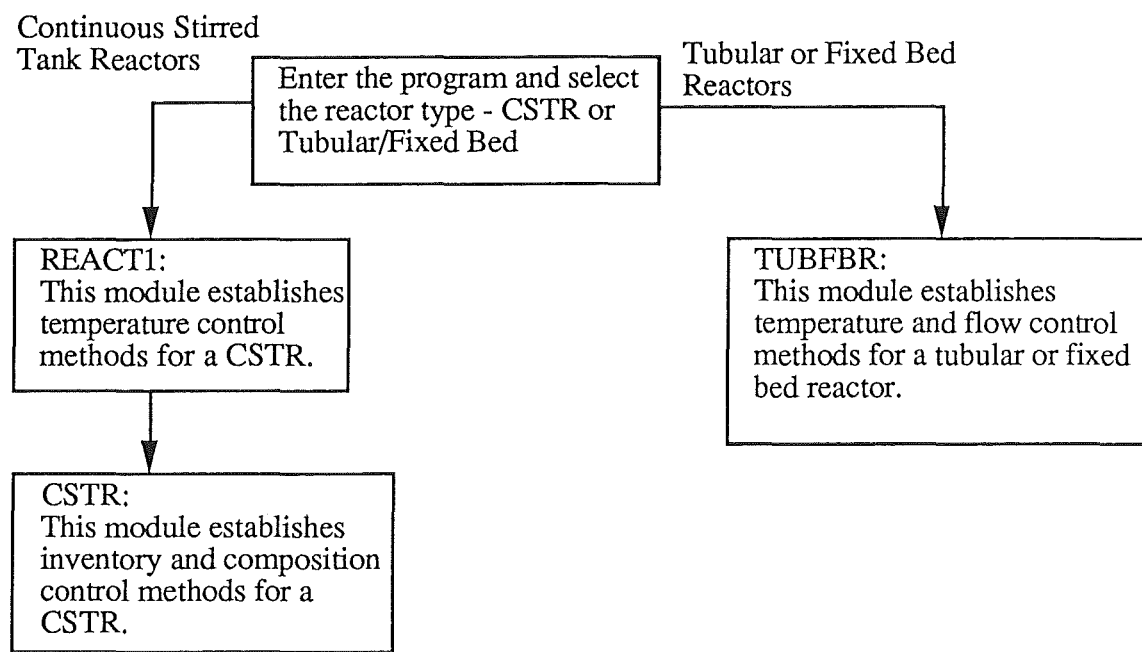


Fig 5. The structure and operation sequence of the reactor control scheme expert system

Many of the control schemes are a direct consequence of the type of equipment chosen for the reactor. As an example, if a boiling liquid reactor with an overhead condenser is selected and it has a vapour product and a nitrogen feed to the condenser there is only one sensible control scheme to use for temperature (Fig. 2, H2). The rules and the input reflect this situation. Many of the input queries are aimed at identifying equipment and the later selection rules acknowledge this key information in recommending control schemes.

6.1 The Modules REACT1 and CSTR

The structure and operation of the modules REACT1 and CSTR is shown in fig. 6. These modules are divided into a number of subgoal sections which are described briefly here. The complete list of rules is included in Appendix AV.

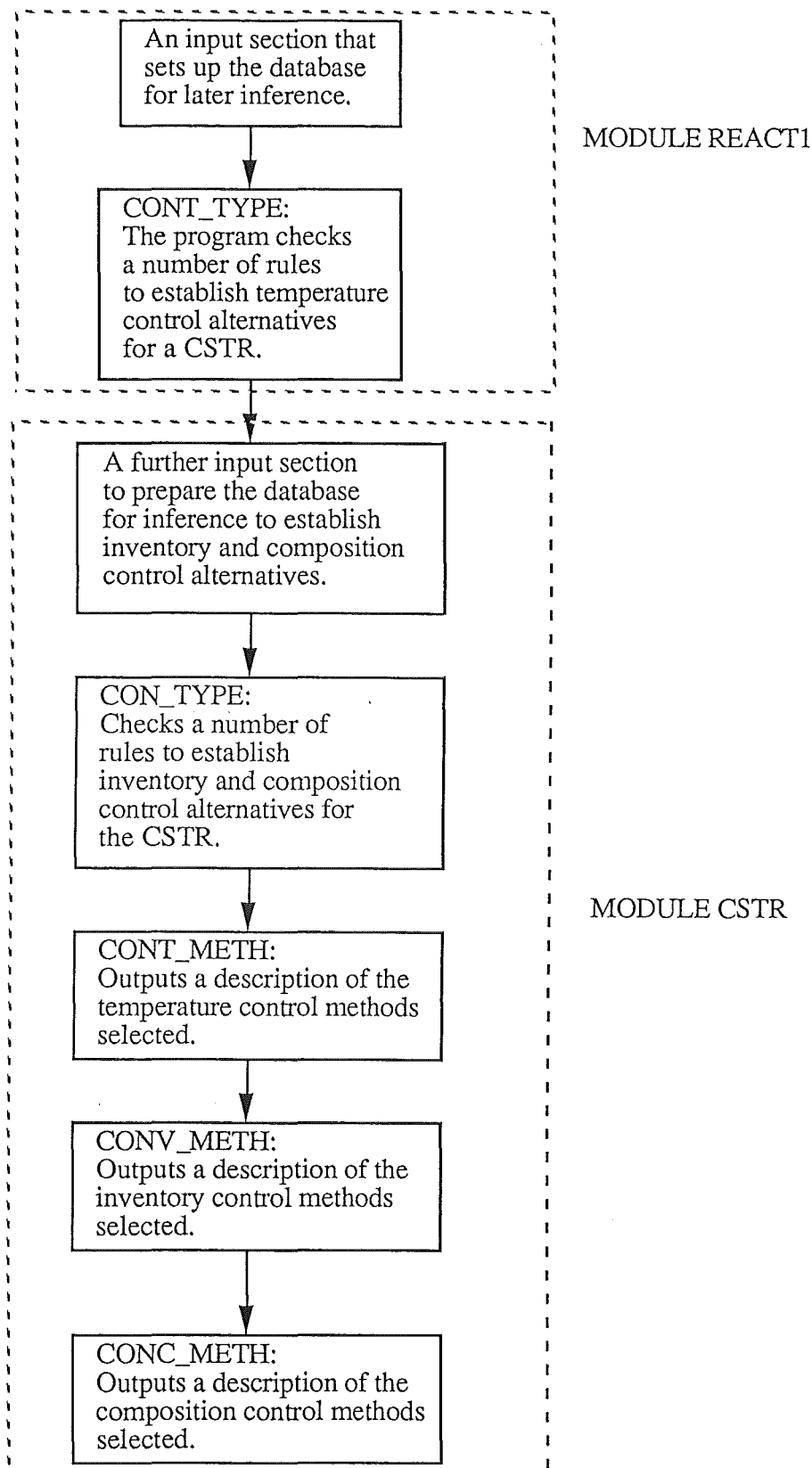


Fig. 6 the structure of the the two modules REACT1 and CSTR for control method selection for Continuous Stirred Tank Reactors

6.1.1 Input

The input sections of the modules REACT1 and CSTR operate in the same way as the interactive input for the distillation program (Chpt 2, Fig. 17). In the first input subgoal in the module REACT1 the program collects information about the reaction, such as whether it is exothermic or endothermic, and about the reactor, for example whether cooling is achieved by a coil or jacket for an exothermic reactor. There are questions about the type of disturbance expected eg. variations in coolant temperature may occur. These can subsequently be remedied by cascading the primary loop onto a secondary coolant temperature control loop.

6.1.2 Reactor Stability

The probable stability of an exothermic reaction plays a significant role in deciding the reactor type and its control system. Although the criteria is essentially a numerical one (section 2.2 in the chapter) as a first approximation a set of heuristic rules was added for assessing stability. These depend on qualitative measures of the heat evolution of a reaction, either small, medium or large. If the heat release is small, or medium without significant lags, it is likely to be stable or stabilised by a feedback loop in the normal low cost equipment, a reactor with a coil or a jacket with once through coolant flow. On the other hand, if a reaction has a medium heat release and the equipment is dominated by deadtime then it is likely to be unstable in a normal closed loop and more expensive cooling arrangements must be used. If the reaction has a large heat release then a boiling liquid reactor should be used.

6.1.3 Temperature Control Method Selection Rules

There are twenty rules for selecting temperature control methods for CSTR's gathered under the subgoal "cont_type". This part of the program has a similar operation to that of the "formloops" subgoal in the distillation program (Chpt 2, Fig 18). The program checks through a number of conditions and concludes that a control type is suitable if these are all satisfied. The rules for exothermic reactors reflect the discussion in section 2.0 of this chapter and will recommend the control approaches described in their appropriate circumstances. An example is;

"If the reactor is cooled using a coil and the reaction is stable and there are no variations in coolant temperature expected then control temperature by throttling the coolant flow through the coil."

This rule accesses the rules for checking stability because a coil can only cool reactions with moderate heat release as it has a limited heat transfer capability. If the

reaction is stable, according to the rules, and the program can identify the reactor as having a cooling coil then the conclusion to use the standard control method of throttling coolant flow is made. There are rules that recommend improvements to the basic configuration. For example, if variations in coolant flow are expected then they should be smoothed out using a secondary coolant flow controller in cascade. There are also rules for catching design errors that mean a control method can't be selected by the program. For example, if a reactor with a coil is specified and the reaction is likely to be unstable then the correct design approach is to change the reactor type and use a jacketed version. This will be recommended by the program. The rule here is;

"If the reactor is cooled by a coil and it is exothermic and unstable then try a reactor with a jacket cooling method instead"

Endothermic reactions are much more easily handled than exothermic ones. The reactor usually runs up against the heat transfer constraint to obtain maximum conversion. Flow control on the heat source and monitoring of the reaction temperature are more appropriate in this instance than a loop between reaction temperature and coolant flow as the valve would run almost wide open any way, unless the plant was turned down in rate. If this does occur the operator can reduce the flow of the heating medium to maintain product composition.

6.1.4 Flow, Inventory and Composition Control in a CSTR

This part of the control system for a CSTR is handled by the module "CSTR" (Fig. 6). The input section establishes facts about the reactor arrangement whether it is single pass or part of a recycle loop, for example, and which variables are available as manipulations.

There is a distinction drawn in the rules between conversion and composition control. In a CSTR the conversion (amount of reactant consumed) is given by the equation;

$$y = \frac{K \frac{V}{F}}{1 + K \frac{V}{F}}$$

The equation shows that if temperature and space-time remain the same ie. careful temperature and level control with unchanging feed flow conversion will also remain constant. This may be adequate for many reactors. However, if exit composition is the control objective and there are changes in inlet composition even with a constant conversion exit composition will alter and further control action is

required to keep it constant. This is possible by varying space-time, reaction temperature (this is only likely to be possible if the reaction has a small heat release) or catalyst addition rate if available.

6.2 Tubular and Fixed Bed Reactors

The final part of the reactor expert system was developed to handle tubular and fixed bed reactors. Its structure and operation are summarised in Fig. 7. A complete listing of the rules is included in Appendix AV.

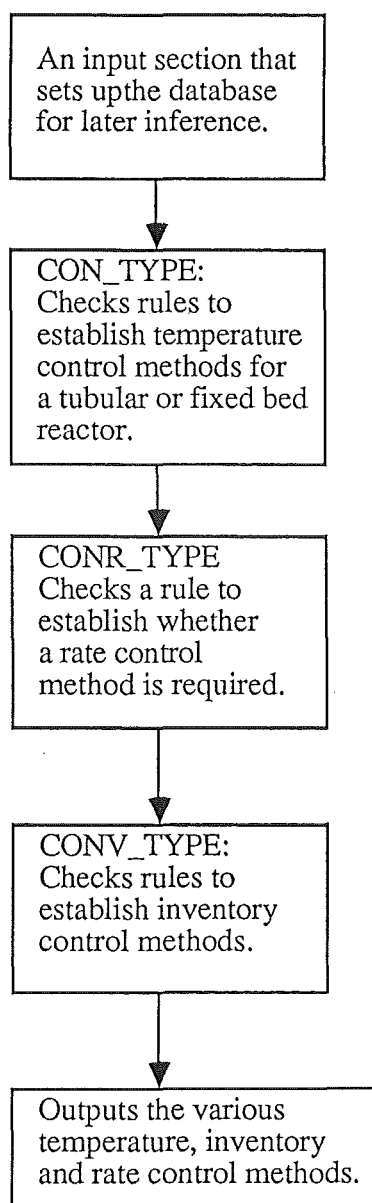


Fig. 7 The structure of the TUBFBR module that recommends control methods for tubular and fixed bed reactors

6.2.1 Input

This section of the program works in the same way as all the previous input sections. Information is required from the user on the type of reaction, heat evolution rate and the reactor type and feed arrangement to the reactor. As tubular and fixed bed reactors are often used in gaseous reactions that occur within a recycle loop, changing pressure and the build-up of inert non-reacting species can affect the reaction so information relevant to these aspects of the designs is also needed.

6.2.2 Stability rules

Similar problems with reactor stability exist for tubular reactors as they did for CSTR's and rules were placed in the program to assess stability. If the heat evolution is small or medium and no significant lags are introduced in the reactor arrangement the reaction is likely to be easily stabilised by a feedback loop.

6.2.3 Temperature Control Methods

Equipment and cooling method play an important part in establishing what kind of temperature control should be used for an exothermic reactor. The likely stability of the reaction governs reactor selection. In this program tubular and heat exchanger reactors with a coolant flowing in the jacket or shell are only applicable if the reaction is easily stabilisable. Tubular or heat exchanger reactors with steam raised in the jacket are suitable for less stable reactions (medium heat evolution in reactors dominated by deadtime) but only heat exchanger types with steam raised in the shell are appropriate for reactions with large heat releases. Therefore there are rules to pick up exceptional circumstances when the wrong type of reactor has been chosen for example;

"if the reaction is exothermic and the heat release is large and the reactor is a tubular type with steam raised in the jacket for cooling then it is unlikely to be stabilisable and a heat exchanger type reactor with steam raised in the jacket should be used"

There are rules to suggest cascade improvements to the control systems and still others to suggest control for fixed bed reactors although there are no stability rules for this class of reactor. There is a rule to recommend reaction rate control which was apparent from studying the Oxo reactor example.

"If a recommended control method is to control temperature by varying jacket steam pressure and a catalyst feed is available as a manipulated variable and the feed

composition or flowrate is variable then control reaction rate (inferred by steam flow from the reactor) by varying catalyst feed to the reactor"

6.2.4 Feed and Pressure Control Methods

There is a set of rules to establish control methods for feed flow and pressure. These identify ratio control methods on feed flowrate and the possibility of controlling pressure by bleed rate from the loop. The rules for handling inerts in a gaseous recycle loop identify whether inerts are fed to the loop and whether their concentration in the feed is variable, for example;

"If the reactor is part of a recycle loop and a purge bleed is available and there are inerts with varying concentration in the feed to the loop and the purge bleed flow is measured then control loop pressure by varying bleed rate and alter the flowrate of feed to the loop in response to changes in purge flowrate"

7.0 The Program at Work

An example is described in this section to illustrate how the program works. The reactor is an exothermic jacketed CSTR using heat transfer fluid circulated through an external heat exchanger for cooling. The reaction has a medium heat release/volume and two liquid feeds. The input session with the program is as follows (user's responses in italics);

"Is the reactor

1) A CSTR

2) A tubular or fixed bed reactor

enter the type [1/2]" - *1*

"Is the reaction

1) Exothermic

2) Endothermic

Enter the choice [1/2]" - *1*

"Is the heat evolution / Volume

1) small

2) medium

3) large

IV-22

Enter the choice [1/2/3]" - 2

" Does the CSTR have As a means of cooling.

- 1) A jacket
- 2) A coil
- 3) No attached heat exchange

Select the option that applies [1/2/3]" - 1

" Is the cooling fluid in the jacket

- 1) Cooling water
- 2) Heat transfer fluid
- 3) Steam generation
- 4) Cooling water or steam for initiating reaction

select the option that applies [1/2/3/4]" - 2

" Are variations in coolant supply temp expected " - y

" Are significant lags introduced by temp measurement, heat removal or reaction mass"
- y

" Is heating required to initiate the reaction" - n

" Are fast dynamics needed in response to temperature fluctuations" - y

" Is the reactor

- 1) A single-pass reactor
- 2) Part of a recycle loop

Enter the type [1/2]" - 1

"Is there feeding the reactor

- 1) A single liquid feed
- 2) Two liquid feeds
- 3) A gas feed and a liquid feed

Select the option that applies [1/2/3]" - 2

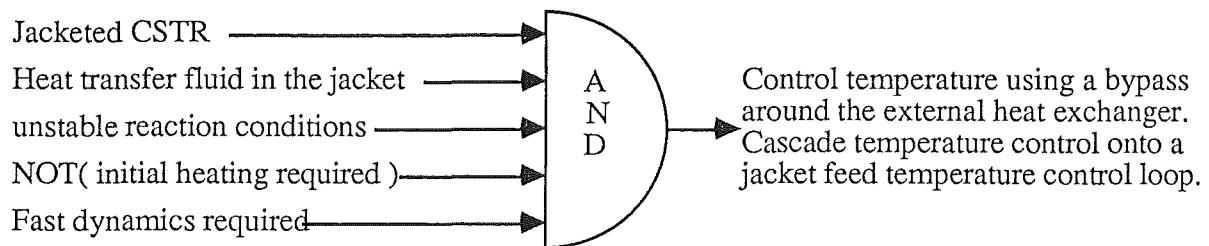
" Is conversion a control objective " - n

" Is exit composition a control objective" - y

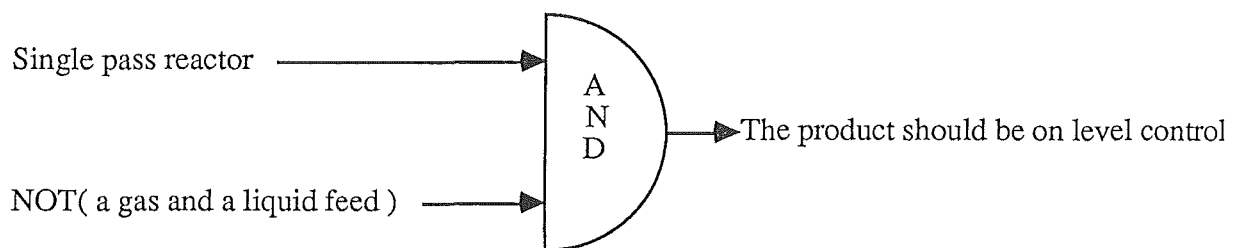
" Are variations expected in reactant feed composition " - n

The expert system concludes that the temperature should be controlled using a bypass around the heat exchanger on the circulating heat transfer fluid and the loop should be cascaded onto a temperature control loop on the coolant entering the jacket. The level should be controlled by the product stream and the two liquid feeds should be flow controlled with a ratio controller between the two streams. The rules that fire to establish the various control types are shown in Fig. 8.

a) Temperature Control Rule



b) Inventory Control Rule



c) Composition Control Rule

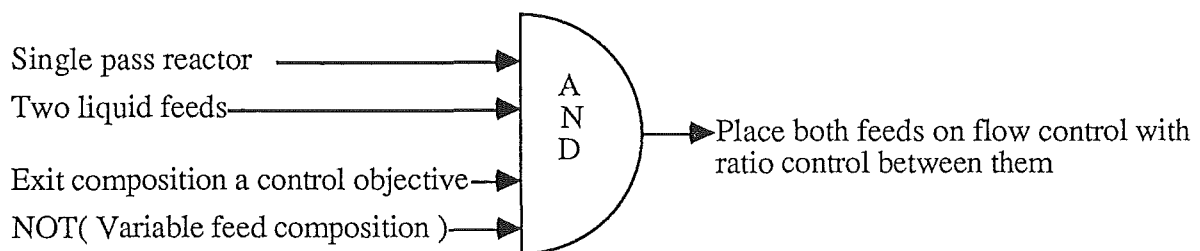


Fig. 8 The rules that fire to establish the control methods for the CSTR example

8.0 Nomenclature

E = Activation energy

IV-24

F = Reactor feedrate

K = Reaction rate constant

R = Universal gas constant

T = Reaction temperature

T_j = Reactor cooling jacket temperature

V = Reaction volume

y = Conversion of reactants in the reactor

Chapter 5 - Structural Controllability

1.0 Introduction

Structural controllability, introduced in the first chapter, was chosen in this work as a suitable method to coordinate between expert systems that synthesize control systems for functional unit (a functional unit is made up of a number of unit operations that perform a complete function, for example the column, condenser, reboiler and reflux drum together make-up a distillation functional unit). As it makes up an important part of a method for whole plant control systems synthesis the concept and its development are explained more fully in this chapter.

2.0 State Controllability

The concept of controllability was first adapted as a technique for control systems synthesis for whole processes by Morari and Stephanopoulos (1980). Their method is based around state controllability and observability

Definition 1- (Kwakernaak and Sivan) The linear system;

$$dx/dt = Ax(t) + Bu(t)$$

$$y = Cx(t)$$

where A is an $n \times n$ matrix, $x(t)$ is an $n \times 1$ vector, B is an $n \times r$ matrix $u(t)$ is an $r \times 1$ vector, y is an $r \times 1$ vector and C is an $r \times n$ matrix, is completely state controllable if and only if it can be transferred from any initial state x_0 to any terminal state $x(t_1)=x_1$ within a finite time t_1-t_0 .

The mathematical test which establishes whether a system meets this criterion is that the controllability matrix

$$P = (B, AB, A^2B, \dots, A^{n-1}B)$$

has full rank (rank n)

Definition 2- The above system is state observable if by observing the output y it is possible to completely define its state at some time before the point of observation. The mathematical criteria to satisfy this property is that the observability matrix

$$Q = (C^T, A^T C^T, \dots, (A^T)^{n-1} C^T)$$

has rank = n

The concept of state observability originates from multivariable state feedback control theory which requires that the system states can be reconstructed from the outputs using an "observer" (a mathematical model relating outputs to states) so that the complete state vector can be fed back to the controller.

State controllability has some significant drawbacks when applied to practical control systems synthesis situations. Some of these were outlined by Morari (1980)- quoting from this paper;

"1) The path going from x_0 to x_1 isn't completely arbitrary. Any practical control input might result in intolerably large deviations before x_1 is reached from x_0 .

2) If the control u is bounded, we might not be able to reach x_1 in the specified amount of time, or not at all.

3) We have no information concerning regulation, ie. what to do when disturbances enter the system.

4) Assuming that we are just interested in keeping a subset of the outputs at their setpoints in the face of disturbances by constructing feedback loops using the available manipulated variables, we cannot deduce if and how this can be achieved.

5) The rank test gives us no quantitative clues as to "how controllable" a system is.

6) The rank test might fail because of some unfortunate parameter choice. In reality, most of the system parameters are determined experimentally and not known exactly. An arbitrarily small variation in some of the parameters might make the system controllable."

State controllability can also be an unnecessarily rigid criterion for a process control system. As an example, to control the pH of the outlet stream from the second of two CSTR's in series it is only necessary to add chemicals to the second

tank. As there is no control on the pH of the first tank this perfectly adequate approach isn't state controllable.

State observability was suggested as the basis of a method to choose the control objectives for chemical plant but the outputs of a practical control system don't need to satisfy this criteria. Morari acknowledged that control objective selection was up to the designer except in the case of non-self regulating states (pure integrators) or unstable states which must always be control objectives.

3.0 Structural Controllability

Morari went on to make a theoretical development of a procedure for control systems synthesis based around state controllability which *could* be used to generate feedback loops and because it uses a structural representation of the equations avoids the pitfalls of the numerical rank tests. This method draws heavily on the concepts of the structural matrix and the cause and effect graph for equation systems;

Definition 3 - A structural matrix is a matrix representation of a differential equation system with entries in the matrix positions that have some value and zeros in all other positions. For a system with two states and a single input the structural matrix representation (A,B) could have this form;

$$\begin{array}{cccc}
 & x_1 & x_2 & u_1 \\
 dx_1/dt & x & x & x \\
 dx_2/dt & & x & x
 \end{array}$$

The justification for using structural representations of process systems is that conclusions can be drawn about controllability from the form of the equations while avoiding the difficulties associated with uncertain or unknown coefficients occurring in numerical modelling.

The generic rank of a structural matrix is the maximum rank that it can have if numeric values replace the non-zero entries. If it is possible to establish an "output set" for a structural matrix (ie assign entries in columns uniquely and completely to all rows) it has full rank. This property is used extensively in structural controllability analysis. It is however possible that a matrix with full generic rank may have numerical analogues that fail a numeric rank test eg.

$$\begin{pmatrix} x & x \\ x & x \end{pmatrix} \text{ has full generic rank but } \begin{pmatrix} 4 & 2 \\ 2 & 1 \end{pmatrix} \text{ has rank} = 1$$

This situation occurs rarely in physical systems however.

It is also possible that an ill-conditioned matrix would pass a rank test but the actual values in such a matrix could lead to real control problems. The condition number and singular values of a matrix have been interpreted as indicators of the extent of this sort of difficulty (this is explained in more detail in Chapter 1).

Definition 4 - The cause and effect graph is a representation of a system of linear differential equations using nodes and edges. For the set of equations

$$dx_1/dt = f(x_1, x_2, u_1)$$

$$dx_2/dt = f(x_2, u_1)$$

the cause and effect graph is;

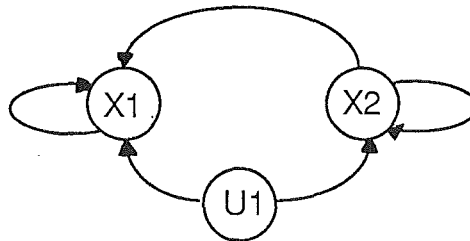


Fig. 1 A cause and effect graph

an edge exists between two nodes if a variable (node 1) appears in the equation describing the differential w. r. t. time of the second node. The edge in this case is directed from node 1 to node 2. This representation of a system is useful for checking for "accessibility" which is important in controllability analysis. A node is said to be accessible from another if a path can be traced in the direction of the arrows from the input node to the tested node. In the graph above x_1 is accessible from u_1 but x_2 is not accessible from x_1 as no path exists between x_1 and x_2 traveling only in the direction of the arrows. A state node must be accessible from an input node for control to be possible.

This representation of the equation system is closely related to the structural matrix. If the matrix has an entry in the i th row and j th column then a directed edge exists in the cause and effect graph between node j and node i .

Structural state controllability theory requires that for a state space system

- 1) Each state node is accessible from at least one input node.

2) The generic rank of the compound matrix (A,B) is n . (the proof of this is available and its sufficiency is assumed adequate in this work (Morari,1977))

The physical interpretation of the requirements for structural controllability can be amplified. If a state node is not accessible from the inputs there is no possible way that it can be influenced by any of them and therefore can not be controlled. Similarly if a state node is not observable (checked by testing the accessibility of the cause and effect graph for (A^T, C^T)) its behaviour can not be deduced from the outputs. This is only important when the behaviour of an unstable state is not transmitted to the control system and action can not then be taken to control it.

The rank test on (A,B) detects pure integrators (non self-regulating states such as tank level for instance) that are not controllable with a given set of manipulated variables. This test is necessary because although all the state nodes are accessible if there are insufficient inputs available to control all the pure integrators in the system it cannot be adequately controlled. This can also be interpreted as a lack of degrees of freedom or dependent rows/columns in the matrix (A,B) . An example taken from Morari demonstrates this;

$$A = \begin{pmatrix} 0 \times 0 \\ 0 \times 0 \\ 0 \times 0 \end{pmatrix}, B = \begin{pmatrix} x \\ x \\ x \end{pmatrix} \quad (A,B) = \begin{pmatrix} 0 \times 0 \times \\ 0 \times 0 \times \\ 0 \times 0 \times \end{pmatrix}$$

In this system there are two pure integrators, x_1 and x_3 . The compound matrix (A,B) has rank = 2 and therefore fails the rank test. Even if one control objective was chosen for this system and paired with the input the system would not be properly controlled because one of the integrators would be left free.

4.0 Structural Controllability Applied to Control Systems Synthesis

In order to adapt controllability theory to the synthesis of control systems for chemical plant Morari changed the state space system representation to reflect Proportional-Integral control. If a system is controlled using SISO loops the control law for each of them is;

$$u = k_c y + \frac{k_c}{T_i} \int y \, dt \quad (1)$$

If the system state vector is augmented with integrals of the outputs the state feedback law becomes $u = K [x \int y]^T$. For a system controlled using PI loops the

coefficients of the matrix K that correspond to the individual SISO loops are defined by equation (1) and the output equation $y = Cx$. The state space representation of a system with the state variables augmented in this way is (here $z = \int y$);

$$\begin{pmatrix} \frac{dx}{dt} \\ \frac{dz}{dt} \end{pmatrix} = \begin{pmatrix} A & 0 \\ C & 0 \end{pmatrix} \begin{pmatrix} x \\ z \end{pmatrix} + \begin{pmatrix} B \\ 0 \end{pmatrix} u$$

If this system can be shown to be controllable Morari concluded it is possible to form a system of PI loops to control the plant. The conditions that must be satisfied to ensure that this is so are;

- 1) (A,B) is controllable.
- 2) The rank of $\begin{pmatrix} A & B \\ C & 0 \end{pmatrix}$ is $n + r$

if condition (2) is satisfied this is sufficient to conclude that the rank of (A,B) is n and rank (A^T, C^T) is also n . Systems satisfying these conditions have been called Integral Control Controllable.

The first stage of Morari's control system synthesis procedure developed from this theoretical basis involves ensuring accessibility within the matrix (A, B) and the setting up of the composite matrix for the system for the rank test required by the second condition. This usually results in there being more available manipulated variables than control objectives and the problem then becomes one of choosing which manipulated variables can be eliminated while still retaining full rank of the composite matrix. Morari suggested two methods for establishing this. The first requires the formation of all distinct output sets for the matrix. Each one of these would be the basis for a regulatory control scheme (the eliminated variables are the ones not required in a set). The amount of computation needed is reduced by reordering the matrix into block form. The reordered matrix has a series of non-zero blocks on the diagonal each with a rank index defined as the number of columns in the block - the number of rows in the block. Morari's shortcut method takes advantage of this reordered form. The elimination information for the submatrix up to and including block k is that as many as

$$\sum_{i=1}^k (\text{Rank Index})_i \text{ columns may be removed}$$

as long as this doesn't violate similar considerations for earlier submatrices. An example is shown in fig 2 along with the elimination information.

$$\begin{pmatrix} \text{x} & \text{x} & \text{x} & \text{o} & \text{o} & \text{o} \\ \text{x} & \text{x} & \text{o} & \text{x} & \text{o} & \text{o} \\ \text{x} & \text{x} & \text{o} & \text{o} & \text{x} & \text{x} \end{pmatrix}$$

Elimination information

Columns	No. that can be removed
1, 2, 3	2
1, 2, 3, 4, 5, 6	3

Fig 2 Elimination information developed for a block form matrix using the shortcut method.

Johnston and Barton (1984) demonstrated that this shortcut method did not produce the correct information in some instances. They developed a more robust algorithm which is explained in more detail later in this chapter.

Morari applied his method to an entire process by setting up the matrix for the whole plant and producing the elimination information for it. This had the disadvantage of producing a large set of manipulated variables which have to be paired with control objectives. The problem is more manageable if the flowsheet is decomposed into subsections and the same technique applied sequentially to each. The solution is then in the form of a sub group of control objectives, a subgroup of manipulated variables and the elimination information for that group. The procedure is fully explained in the original reference (Morari and Stephanopoulos, 1980)

4.1 Improvements to Structural Controllability Methods

This early work by Morari and Stephanopoulos (1980) was developed by other researchers. Johnston and Barton (1985a) furthered the understanding of the physical meaning of the rank test for Integral Control Controllability (ICC) and improved some aspects of the synthesis procedure.

4.1.1 Physical Meaning of The Controllability Rank Test

Johnston and Barton demonstrated that as well as detecting pure integrators that cannot be controlled with a given set of manipulated variables the rank test also exposed three further difficulties in a system;

- i) Contractions in the cause and effect relationships between manipulated variables (inputs) and outputs;

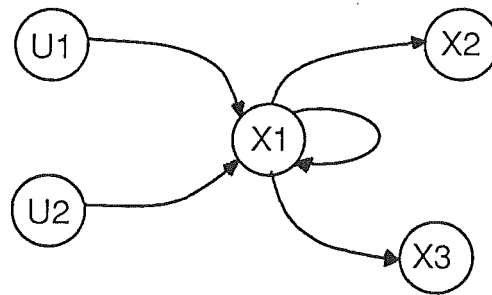


Fig. 3 A contraction in a cause and effect graph

The two manipulated variables influence the two controlled variables through a single state - clearly control is impossible in this situation. The often quoted three tank arrangement is a physical example of a contraction (Morari and Stephanopoulos, 1980, Russell and Perkins, 1987).

ii) Lack of access. This is an accessibility problem as described previously. In the example shown in Fig. 4 there is no access to state x_3 from either u_1 or u_2 . This would be a problem if x_3 was an integrator or unstable state.

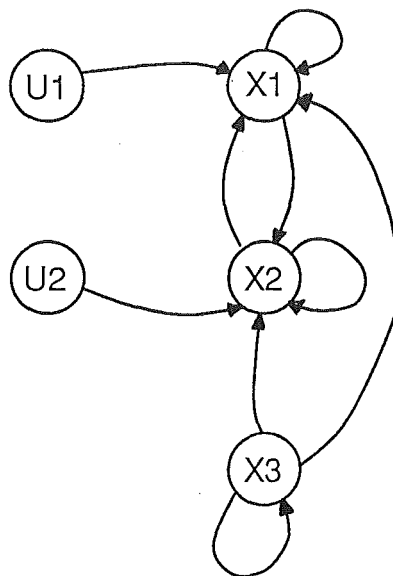


Fig. 4 Lack of access in a cause and effect graph

iii) Absence of direct access. The only access by manipulated variables to some states is via other states which are themselves specified as outputs. As these states are technically invariant under good control then this becomes an unfeasible configuration;

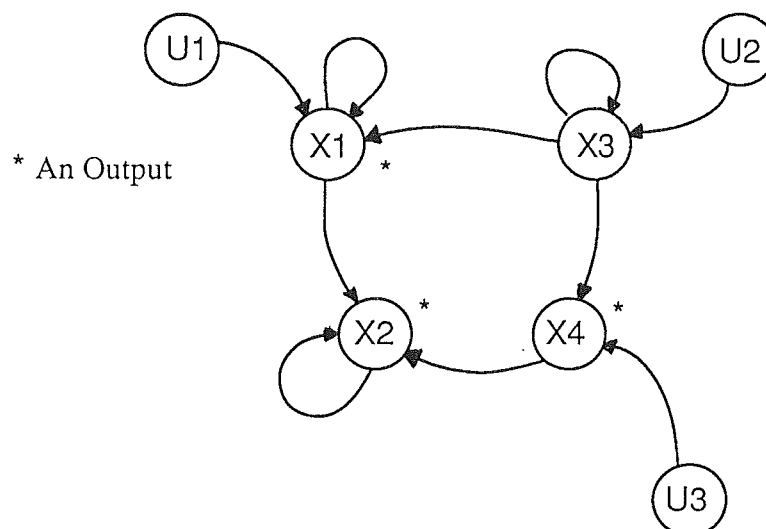


Fig. 5 A cause and effect graph showing lack of direct access

The Johnston and Barton algorithm for finding generic rank is based around the detection of a set of K rows in the structural matrix that have entries in less than K distinct columns. In this case it is impossible to form an output set for the matrix and it can therefore never attain full rank. This deficiency is called a "dilation". All the above physical limitations to control appear as a dilation in the matrix and are detected by this method. They were able to apply this algorithm in a modified form to provide the required information on which manipulated variables can be left out while retaining full rank in a matrix (Johnston et al, 1984). This is claimed with very little evidence to be both less computationally demanding than repeated formation of output sets and more robust than the short-cut technique developed by Morari. Programs were written in FORTRAN at first and then in PROLOG based on Johnston's work to apply the algorithm and obtain elimination details. An example matrix and the elimination information for it are shown in Fig. 6.

$$\begin{array}{c}
 1 \ 2 \ 3 \ 4 \ 5 \\
 \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \end{matrix} \left(\begin{array}{ccccc}
 & x & x & & \\
 & & x & x & \\
 x & & & x & \\
 & & x & x & x & x
 \end{array} \right)
 \end{array}$$

Columns	No. of columns which can be eliminated
1, 2, 3	0
4, 5	1

Fig. 6 A structural matrix and elimination details to retain full generic rank

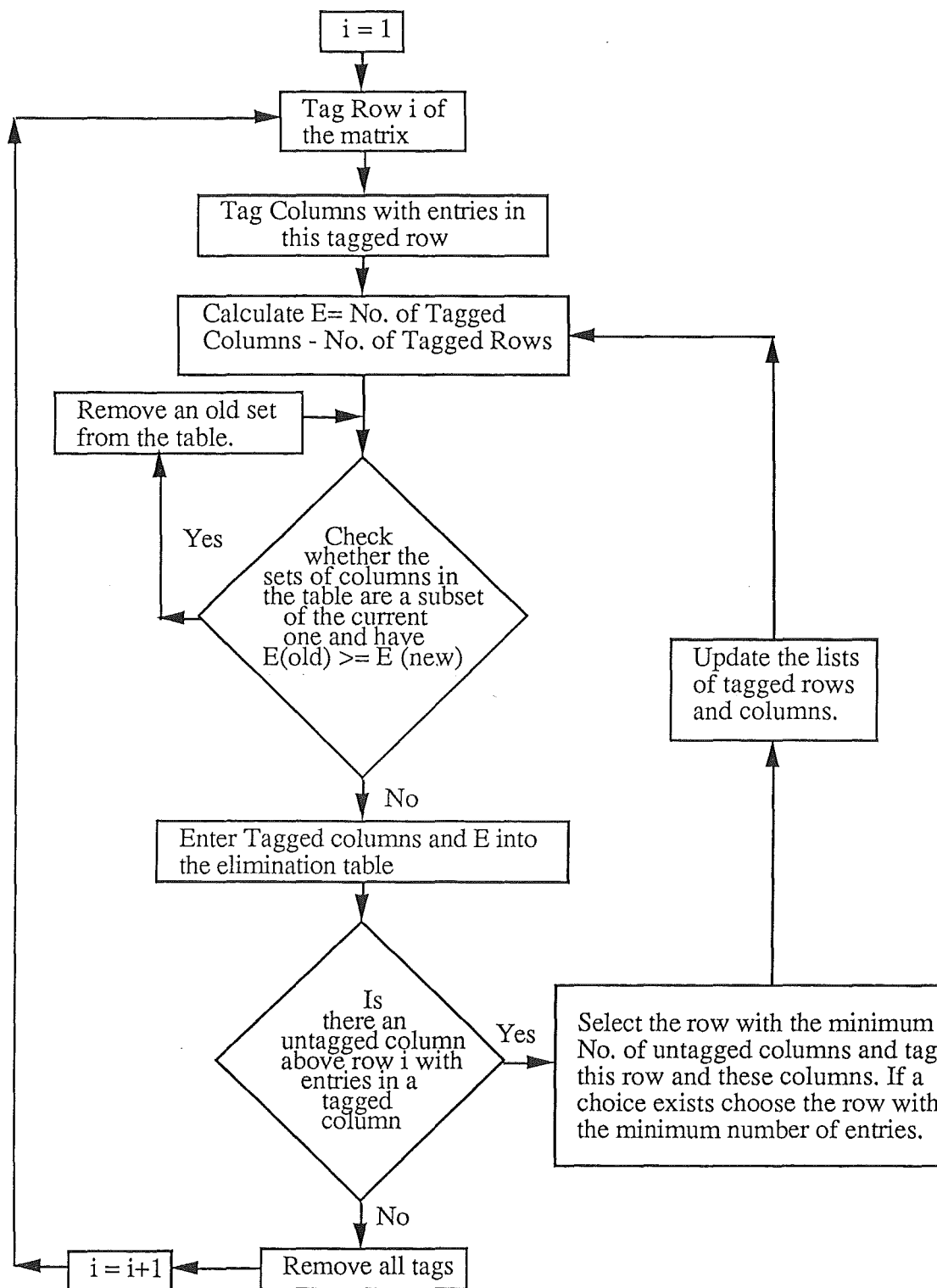


Fig. 7 A flow diagram of the PROLOG excess column elimination program

A flow diagram for the PROLOG version is shown in Fig. 7 and full details of the method can be found in Johnston et al, 1985a . The programs written for this

work produced the same information, on a number of examples, as was originally published.

4.1.2 The COORDINATOR Matrix

The COORDINATOR matrix was developed by Johnston and Barton to implement Morari's synthesis procedure when applied unit by unit to whole plants non-iteratively. It is a matrix with all the plant control objectives as rows and the manipulated variables as columns. There is an entry in a matrix position if a "direct access path" exists between a manipulated variable and a control objective in the composite matrices for the unit operations that make up the plant. A direct access path is a path between input and output that satisfies the accessibility rules and which does not pass through a state that is an output. If the COORDINATOR matrix has full rank it is possible to form at least one output set ie. a one to one correspondence between control objectives and manipulated variables. This means that control loops can be formed between a subset of the manipulated variables available and all the required control objectives. It is still possible in this situation that problems may exist at a unit level because of a contraction (see (i) above in "physical meaning of the controllability test")- and the system may not be controllable. This has to be checked by ensuring that unit operation control systems pass the controllability tests. The elimination algorithm developed by these workers can be applied to the COORDINATOR matrix, reordered into a block structure using the method developed by Morari and subsequently modified by Johnston and Barton, to obtain information about which manipulated variables can be left out on a plant level while still retaining the generic rank of the COORDINATOR. It is this extra information which makes the procedure non-iterative when compared with Morari's original method.

5.0 Functional Controllability

Perkins and Russell (1987 and 1988) considered that functional or output controllability was a more appropriate test than state controllability for control systems synthesis. Functional controllability requires that for a system with input $u(t)$, state $x(t)$ and output $y(t)$ for any given output trajectory ($y(t): t > 0$) which is zero for $t < 0$ there exists an input trajectory $u(t)$ which generates $y(t)$. This requirement would be sufficient for a control system to hold a set of outputs at their steady state values (representing a system trajectory) usually all that is necessary for effective control. An equivalent requirement is that the transfer function relating inputs to outputs ie. $y(s) = G(s)u(s)$ has an inverse. For a linear system this can be shown to be equivalent to requiring that the composite matrix with the same form as previously discussed;

$$\begin{pmatrix} A & B \\ C & 0 \end{pmatrix} \text{ has full rank}$$

The only difference between Integral Control Controllability and Functional Controllability (FC) is that a FC system does not have to satisfy accessibility criteria and therefore some unstable state in the plant might not be specified as an output and escape control. This may be seen as a weakness in the method, but in fact all design engineers would include such variables as control objectives.

The other significant contribution by these researchers was to extend work on structural controllability to define a test which could be applied to differential algebraic equation (DAE) models of physical systems. This makes the algebraic manipulation required to reduce DAE models to the standard linear systems form redundant when controllability is being considered and dovetails well with work on the equation-oriented flow sheeting package SPEED-UP.

The Perkins/Russell controllability tests for DAE systems are rank tests on structural matrices developed from the DAE representation of a system ie.

$$F(t, X', X, Z(t), U(t)) = 0$$

$$Y(t) = M \begin{pmatrix} X' \\ X \\ Z \end{pmatrix}$$

X is the vector of system states

X' is the vector of state derivatives

Z is a vector of algebraic variables

U is a vector of system inputs

t is time

The inner system matrix

$$PI = \begin{pmatrix} \frac{\partial F}{\partial \dot{X}} & \frac{\partial F}{\partial X} & \frac{\partial F}{\partial Z} \\ I & -sI & 0 \end{pmatrix}$$

must have full rank as must the expanded system matrix

$$PE = \begin{pmatrix} PI & \frac{\partial F}{\partial u} \\ M & 0 \end{pmatrix}$$

Perkins also suggested that for a feasible SISO control loop to be formed between a manipulated variable and a control objective a path must exist from the column representing the manipulated variable to the row representing the control objective - this is analogous to a path through a cause and effect diagram defining accessibility. This condition is sufficient for a finite gain to exist between input and output.

5.1 Perkins/Russell Rank Finding Algorithm

This work is based around Duff's algorithm (Duff, 1981) for finding an output set and therefore confirming the generic rank of a structural matrix. It works by forming "augmenting paths" through the matrix structure. These are related to paths showing accessibility in cause and effect graphs when the augmenting paths are used to assign columns in the expanded system matrix (PE) to form an output set starting from the basis of an assignment made on a reordered inner system matrix (PI). Experiments on examples from the literature show that Duff's algorithm provides similar elimination information to the Johnston and Barton approach in some linear system cases but not all (Appendix AII).

6.0 Conclusion

The Perkins and Russell tests would have to be used if a system could only be modeled using a DAE set and couldn't be reduced to a system of linear differential equations. This might occur for some high index problems. Their work, and a consideration of the physical meaning of rank test failure, suggests that this test indicates a more fundamental control problem than the inability to form PI loops. Systems which fail this test cannot be adequately controlled by systems using any control law.

Programs based on the rank finding algorithm developed by Johnston and Barton were used in this work because methods previously proposed in the literature were shown to be erroneous or did not provide information in the appropriate form to organise the unit operation subroutines. Their concept of a COORDINATOR matrix and the manipulated variable elimination information developed from it could be used

to allow knowledge based systems for control systems synthesis to be applied sensibly to a complete flowsheet.

7.0 Nomenclature

$x(t)$ = The vector of time variant system states

$u(t)$ = The vector of time variant system inputs

y = The vector of system outputs

K_c = The proportional gain for a single-input single-output controller

T_i = The integral time for a single-input single-output controller

K = The matrix of controller gains

t = time

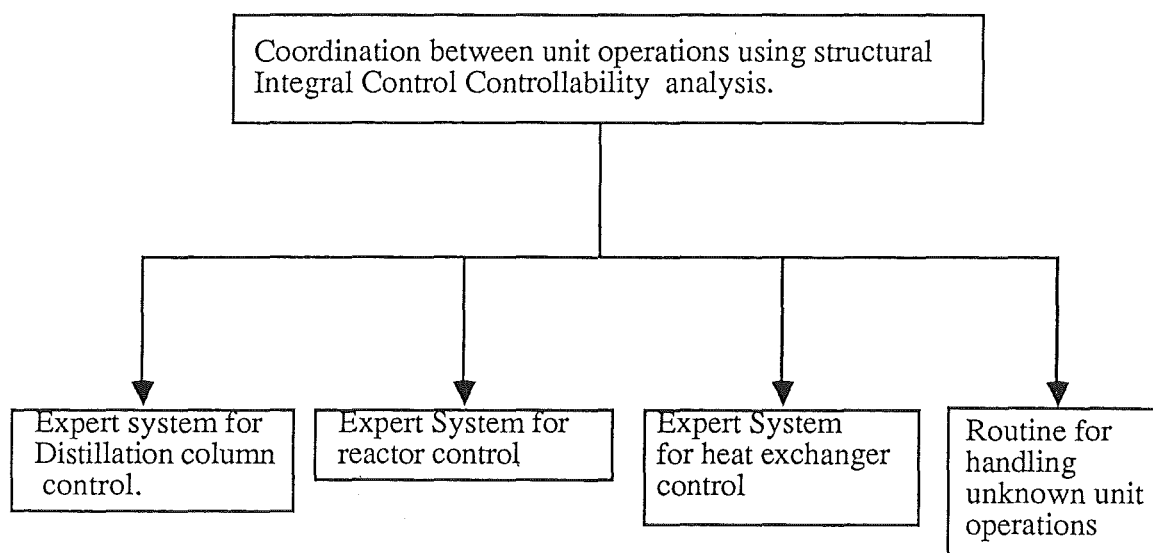
Chapter 6 - Whole Plant Control System Synthesis

1.0 Whole Plant Control System Synthesis Using Expert Systems

Whole plant control systems synthesis has proved difficult using theoretical methods alone because of the complex mix of factors that have to be taken into account. Design information, outside of the control sphere, has a significant effect on control decisions. The choice of equipment often decides the control system. For example, the different types of cooling system for exothermic reactors described in Chapter 4 have particular control configurations. The program should be able to assess whether the equipment choice is correct and suggest the matching control system. Other types of design facts are also important. For instance, in heat exchanger control if cooling water temperature is likely to exceed 50°C when its flow is reduced heat exchanger scaling will increase and this rules out the option of throttling cooling water for temperature control. An expert system, equipped with this kind of knowledge, can quickly narrow down a large number of initial possibilities into a smaller and more realistic group. The time to use theoretical techniques or dynamic modelling is after heuristic arguments and steady-state calculations have ruled out many of the original possibilities. At this point as well, an expert system is useful to direct the user to, or even call as a subroutine, the technique needed to judge the best configuration. For example, the use of the relative gain array by the distillation expert system to help choose between dual composition control options for distillation columns. With these arguments in mind the unit operations expert systems were integrated with structural controllability to form a prototype synthesis program. The program structure is shown in Fig. 1.

Some form of coordination between the control systems recommended for unit operations during synthesis must be supplied to prevent the use of a manipulated variable that is already paired with a control objective in another unit operation or to prevent the use of a manipulated variable in one unit operation when it must be paired with a control objective in another system or unit operation. This occurs, for example, if a control objective is affected by only one manipulated variable which may in turn

affect others. Integral Control or Functional Controllability tests perform this task and also guarantee that a process control system meets a controllability criterion. The outputs of a process control system satisfying functional controllability will follow a required trajectory for certain values of the inputs. As was discussed in the previous chapter this is an important property for a process control system to have especially when its function is to hold the outputs at their setpoints.



Other unit operations could be added at this operating level

Fig. 1 The structure of the whole plant control systems package.

Structural controllability requirements are not sufficient to completely synthesize a control system. If some of the manipulated variables can be omitted from the control system, if improvements (cascade or feed forward) need to be added or if any other control idiom is required for start-up or shutdown the approach on its own can't help. Expert systems can provide this extra dimension.

2.0 Preliminary Control System Synthesis Strategies

Before using a package of the type described above some initial steps must be taken to shape the problem into the right form. The control objectives for the process must be established and appropriate measurements that reflect these objectives selected. There are already methods available to handle this aspect of the problem and some of them were mentioned in the introductory chapter. Fisher and Douglas's (1988c) approximate methods or the more rigorous work by Morari and Stephanopoulos (1980) are possibilities here.

A preliminary synthesis strategy which follows the path suggested by Morari and Stephanopoulos and subsequently developed by Johnston and Barton is used here and the discussion below is based on the work of the second set of authors. Structural matrices are set up for the unit operations that make up the plant and checked to see that unit level controllability checks are satisfied. A COORDINATOR matrix of the control objectives, the rows of the matrix, and manipulated variables in the process, the columns of the matrix, is then constructed. If a manipulation affects a control objective there is an entry at that point in the matrix. Before the synthesis can continue this matrix must have full rank. If the problem has been overspecified then some of the less important control objectives must be omitted from the control system. The elimination algorithm suggested by Johnston and Barton, run on the transposed COORDINATOR matrix, pin-points groups of control objectives and the number that must be removed to cure the singularity in the matrix. The COORDINATOR matrix at this point has full rank. However the more likely scenario is that there will be more manipulated variables than are required. If the elimination algorithm is run on the COORDINATOR the elimination information for the process is found. The manipulated variables, the matrix columns, are split up into groups and the number of columns within each group that can be left out of the matrix and still leave the reduced-in-size matrix with full rank are identified. Examples of matrix elimination information are included in Chapter 5 and later in this chapter. If, in the formation of the regulatory control system, not enough manipulated variables from a group are used then it will be impossible for all the control objectives to be paired and the process would no longer be controllable. At this stage the expert systems can be used to provide the extra dimension required to the synthesis.

3.0 The Control Systems Synthesis Package

The most significant problem in the development of an overall package is to write a routine to integrate the control systems expert system with the elimination information for the plant. The structure of this program is shown in Fig. 2.

The routine picks out a unit operation identified as a heat exchanger, distillation column, reactor or unknown unit operation (one that is not yet identified with an expert system) using a calculation order established by the designer and entered into a database for the problem. In the usual calculation order the furthest downstream unit operation from the feed is the first and earlier unit operations are sequentially numbered back through the flowsheet until the first unit operation becomes the final one in the order. Other orders could be used at the discretion of the designer. The program also has in the database a list of available manipulated variables

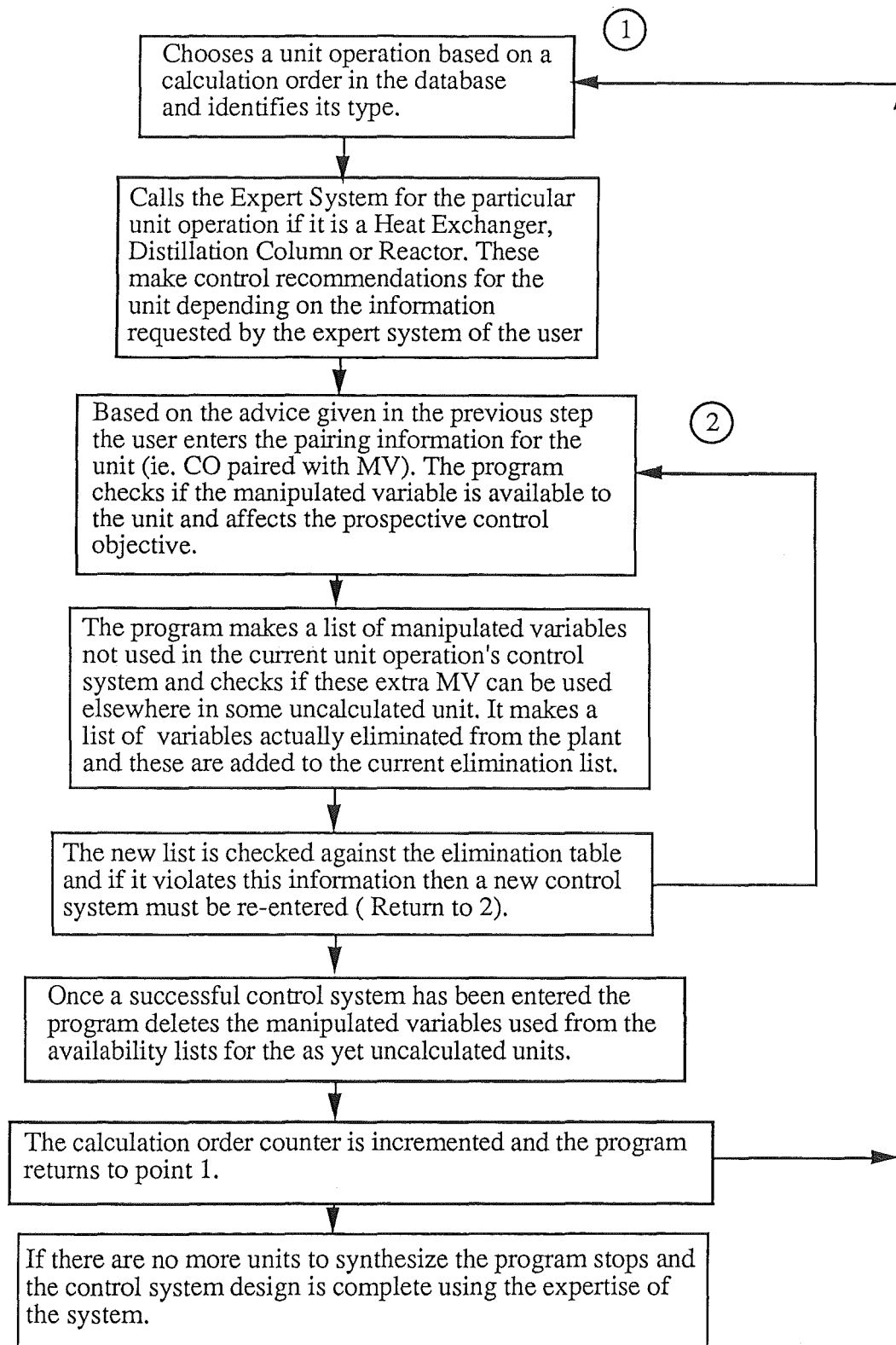


Fig. 2 Structure of the controlling routine for the whole plant control system synthesis program

and control objectives for each unit operation and the elimination information for the plant COORDINATOR.

The controlling routine calls the appropriate expert system which makes control system recommendations. The expert systems have now become subroutines but they work in the same way as the stand alone versions with only minor changes in code to allow them to be integrated into the larger package. The user then makes a selection from the suggested control systems following the general philosophy that the final design decisions should remain with the designer rather than program.

At this stage the user enters the pairing for the selected control system. The program forms a list of the manipulated variables chosen for the unit operation and compares it with the list of possibilities in order to identify manipulated variables available to but not used in the local control system. If these variables can't be used in an as yet uncalculated unit (they are not a member of the appropriate manipulated variable list) then they will not form part of the final control system and are added to the eliminated variables list resident in the database. This list is checked against the elimination information for the COORDINATOR. If a subset of the list belongs to a group identified in the elimination table and the size of the subset exceeds the number from that group which can be left out of the final process control system a new local control system must be entered. For example, if the list contains a variable belonging to a group that must all be used somewhere in the control system it would violate the elimination information. There must be at least one successful control pairing for the current unit because the COORDINATOR matrix has full rank.

After a successful control system is established for the current unit operation the program removes the used manipulated variables from the lists of variables available to other as yet uncalculated units. This ensures that variables can't be used twice in different control loops in separate pieces of equipment.

The calculation order counter is incremented and the program retrieves the next unit and repeats the process by calling the appropriate expert system. After the last unit is calculated the program stops.

In order that the program can handle complete flowsheets, even though expert systems are available only for heat exchangers, distillation columns and reactors, a further routine was added to the program. This is called the "unknown unit" module and it provides possible pairing information based strictly on the COORDINATOR controllability criteria ie. the manipulated variable must affect the control objective and not be paired up with more than one. This is equivalent to finding all the output sets for the submatrix in the COORDINATOR that applies to that unit. This is a minimum

amount of information and requires more input from the designer than if an expert system was available. The structure of this routine is shown in Fig. 3.

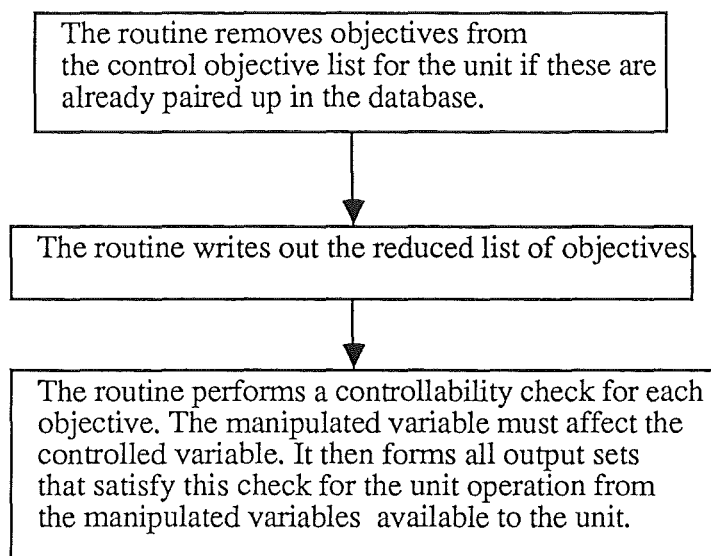
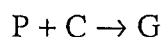
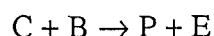
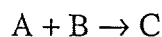


Fig. 3 The operation of the "Unknown Unit" routine

4.0 A Control System For a Complete Process

The program can be applied to an example flowsheet as a demonstration of its usefulness. The flowsheet (Fig. 4) is taken from Johnston, Barton and Brisk's work (1985b) and is a modified version of the Williams-Otto plant, well known as a test plant for dynamic studies (Williams and Otto, 1960). The later authors have added an extra distillation column (Column 2), a second reactor (Reactor 2) and another decanter (Decanter 2) but the same six organic species (A, B, C, E, G, P) are involved. Physical properties and reaction kinetics are the same in both the original and the revamped plants.

Pure liquid streams of A and B plus a recycle stream (F₁₇) which is 98% w/w B feed Reactor 1, a water cooled CSTR operating at 350 K. Three exothermic reactions take place;



C and E are reaction intermediates with a lesser value than the primary product P. G is an oily byproduct that has no real value but must be separated from the other products.

The reactor effluent stream (F_1) is cooled to 311 K in Heat Exchanger 1 because this effectively stops the reactions proceeding and also causes the stream to separate into two phases. The heavier phase (F_4) is 50 % w/w B and 50 % w/w G and there is a complete separation of G from the lighter phase. In the original Williams-Otto plant the heavy phase was 100 % G. Decanter 1 is sized to allow sufficient residence time for the two phases to separate cleanly. The light phase (F_3) is fed to Reactor 2 along with the the bottoms stream (F_{14}) from Column 1. This additional reactor operating at 360 K allows a higher conversion of feedstock to products. The same three reactions occur but because the heat released is significantly less than that required to preheat the feed to reaction temperature at normal rates the reactor must be heated with steam. This is unusual for an exothermic reaction. The product from Reactor 2 provides reboil heat to Column 2 and is then cooled further in Heat Exchanger 2 to 311 K again to allow separation into two phases and the removal of G from the lighter phase which occurs in Decanter 2. The light phase from Decanter 2 (F_9) feeds Column 1 which separates the primary plant product, P, from the other species. The top product (F_{11}) is 98 % w/w P with 2 % A as an impurity. The bottoms stream is split into two. Some leaves the plant as a secondary product (F_{13}), perhaps for further processing, while the rest is recycled to Reactor 2. The heavy phases from the two decanters (F_4 and F_{10}), both of the same composition, are combined and feed Column 2 which distils unreacted B from G. The top product (F_{16}) is 98 % B which, after being heated in Heat Exchanger 3 is recycled to Reactor 1. The bottom product (F_{19}), essentially G, is cooled in Heat Exchanger 3 and leaves the plant as a waste stream.

The plant flowrates required to produce 20000 tonnes per annum of 98 % P are shown in Table 1 (Johnston, 1985). The product compositions from the two reactors were confirmed using the process simulator SPEEDUP assuming the reaction kinetics from the original Williams and Otto paper still applied. In this paper the plant is described as being subject to 3 types of disturbance. The most significant is a cycling in the flowrate of the feed stream of A which may increase or decrease by up to 950 kgs per hour on a regular basis. The temperatures of the feed streams fluctuate a little around 294 K and the cooling water has both a daily and an annual cycle in temperature.

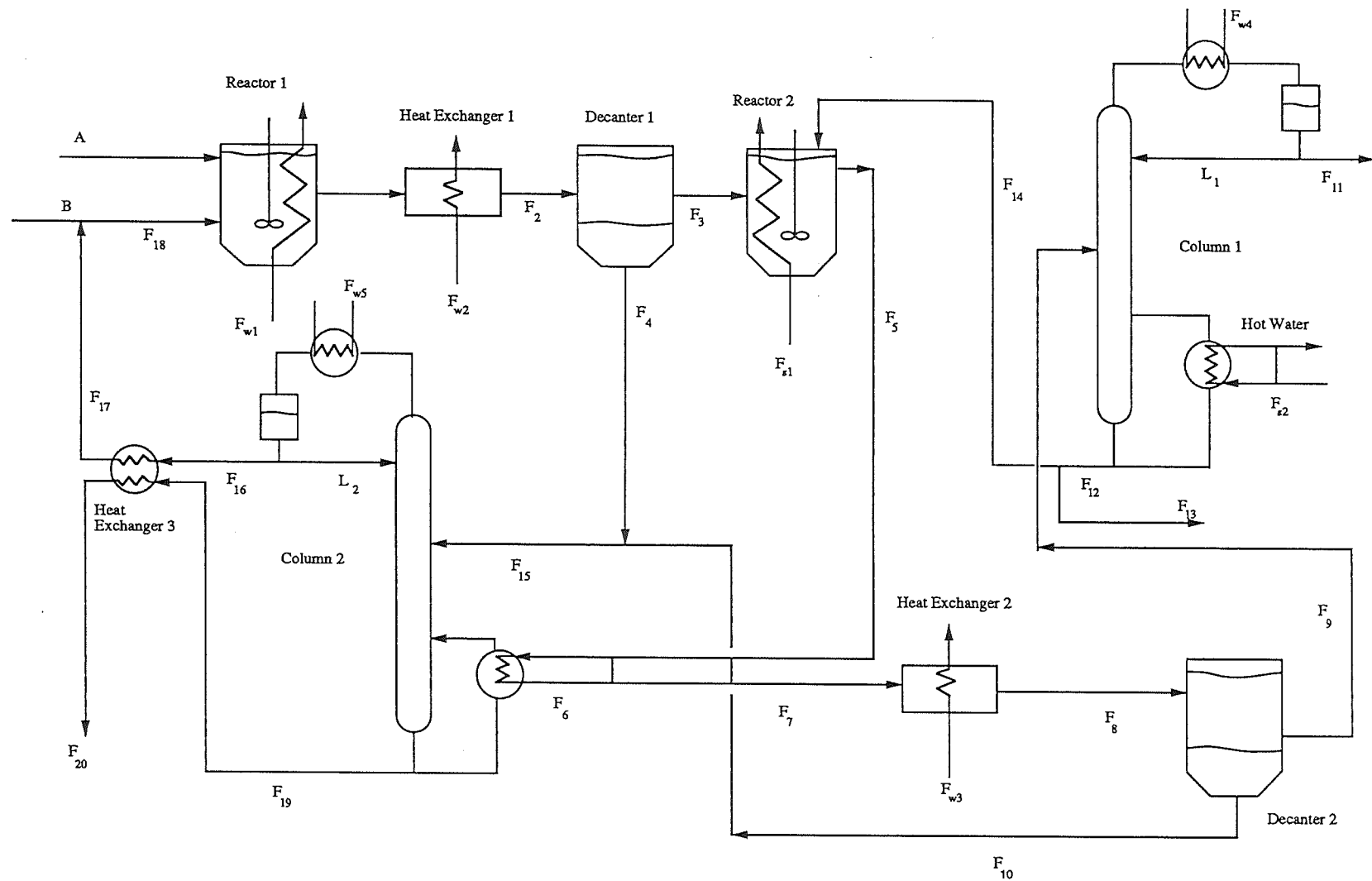


Fig. 4 Test plant for control systems synthesis (modified Williams-Otto plant)

Table 1. Steady-state stream data for the test plant

Stream	F _a	F _b	F ₁	F ₃	F ₄	F ₅
Flow (Kg/hr)	6576.0	14000.0	23762.2	19950.0	3812.2	37660.9
Mass Fractions						
A	1.0	0.0	0.1008	0.1201	0.0	0.0518
B	0.0	1.0	0.4054	0.3873	0.5	0.2523
C	0.0	0.0	0.0215	0.0256	0.0	0.0098
E	0.0	0.0	0.2786	0.3319	0.0	0.5324
G	0.0	0.0	0.0802	0.0	0.5	0.0367
P	0.0	0.0	0.1135	0.1351	0.0	0.1170
Temp (K)	294.3	294.3	350.0	311.0	311.0	360.0

Stream	F ₆	F ₉	F ₁₀	F ₁₁	F ₁₃	F ₁₄
Flow (Kg/hr)	33732.0	34899.4	2761.5	2697.9	14490.7	17710.8
Mass Fractions						
A	0.0518	0.0559	0.0	0.02	0.0589	0.0589
B	0.2523	0.2327	0.5	0.0	0.2522	0.2522
C	0.0098	0.0106	0.0	0.0	0.0115	0.0115
E	0.5324	0.5745	0.0	0.0	0.6226	0.6226
G	0.0367	0.0	0.5	0.0	0.0	0.0
P	0.1170	0.1263	0.0	0.98	0.0547	0.0547
Temp (K)	340.0	311.0	311.0	306.5	311.0	311.0

Stream	F ₁₅	F ₁₆	F ₁₈	F ₂₀
Flow (Kg/hr)	6573.7	3186.2	17186.2	3387.5
Mass Fractions				
A	0.0	0.0	0.0	0.0
B	0.5	0.98	0.9963	0.0485
C	0.0	0.0	0.0	0.0
E	0.0	0.0	0.0	0.0
G	0.5	0.02	0.0037	0.9515
P	0.0	0.0	0.0	0.0
Temp (K)	311.0	305.0	299.1	330.0

The paper by Johnston et al on this problem demonstrated some preliminary work including selection of controlled variables and elimination of excess control objectives before arriving at a COORDINATOR matrix with full rank, 21 controlled objectives by 23 manipulated variables. Therefore two of the manipulated variables can be excluded from the control system if SISO loops are formed.

4.1 Deficiencies in a Solution Using Structural Controllability Alone

The elimination algorithm was applied to this matrix and the elimination information for the matrix developed. The COORDINATOR and its elimination data are shown in fig. 5.

	F_{18}^p	F_1	F_{w2}	F_4	F_3	F_{14}	F_5	F_{w3}	F_{10}	F_9	F_{s1}	F_8	F_{16}	F_{s2}	F_{w4}	L_1	F_{11}	F_{12}	L_2	F_6	F_{w5}	F_{19}	F_{w1}
F_A/F_{18}	x																						
STR1	x	x																					
T_2			x	x																			
V_{G1}			x		x																		
V_{E1}			x		x	x																	
F_3/F_{14}						x	x																
STR2						x	x	x															
T_8								x															
V_{G2}								x		x													
V_{E2}								x		x													
T_5					x	x	x				x												
F_{18}^p												x	x										
P_{c1}														x	x								
C_{11P}										x				x		x	(x)						
H_{rd1}														x	x		x						
H_{spl}									x						x			x					
C_{16B}												(x)							x	x			
P_{c2}																				x	x		
H_{rd2}													x						x		x		
H_{sp2}																			x	x		x	
T_1	x	x											x									x	x

Extra entries not present in the original. (x)

The COORDINATOR matrix for the example process

Manipulated Variables	No. that can be eliminated
$FP_{18}, F_1, F_{w2}, F_3, F_4, F_{14}, F_5, F_{w3}, F_{10}, F_9, F_{s1}$	0
$F_{s2}, F_{w4}, L_1, F_{11}, F_{12}$	1
$F_b, F_{16}, L_2, F_6, F_{w5}, F_{19}, F_{w1}$	1

Elimination information for the matrix

Fig. 5 COORDINATOR matrix and its elimination data for the example process

At this point the original authors found a basic regulatory control system for the plant by excluding variables that did not violate the elimination conditions without

giving reasons why particular ones were left out. Their solution is shown in Fig. 6. There was very little choice then possible in the final pairing of variables which they emphasize by rewriting the COORDINATOR with the eliminated variables removed and suggesting it be used as a guide to pairing variables. Their rewritten COORDINATOR shows just two alternative solutions around Column 2 and the pairing for the rest of the control objectives is determined. This approach totally obscures the fact that there are actually 91 different solutions to the problem of pairing up the 21 control objectives that satisfy structural controllability requirements. This was discovered by forming all the output sets for the process using the routine written for unknown unit operations. The control objectives making up the first 11 rows of the COORDINATOR matrix in Fig. 5 must be paired with the manipulated variables which have an entry on the diagonal in the appropriate row. These 11 manipulated variables comprise the first group in the elimination table for the plant and none of them can be left out of the final control system. There are 10 further control objectives to pair with 12 manipulated variables to complete the basic regulatory structure. The alternatives arise in two parts of the plant. Around column 1 there are five manipulated variables and only four control objectives so one can be omitted (this is the second group in the elimination table of Fig. 5). Seven configurations, listed in Table 2, are possible. The type of composition control for each configuration is identified as well as any difficult loops.

		Control Objectives				Comments
		P_{c1}	C_{11p}	H_{rd1}	H_{sp1}	
Manipulated Variables	1	F_{s2}	L_1	F_{w4}	F_{12}	V/F, cooling rate - level
	2	F_{w4}	F_{s2}	L_1	F_{12}	V/F composition control
	3	F_{s2}	L_1	F_{11}	F_{12}	Direct composition control
	4	F_{w4}	F_{s2}	F_{11}	L_1	Reflux - column level loop
	5	F_{w4}	L_1	F_{11}	F_{s2}	V/F composition control
	6	F_{w4}	F_{s2}	F_{11}	F_{12}	Direct composition control
	7	F_{w4}	L_1	F_{11}	F_{12}	Direct composition control

Table 2 Alternative Control arrangements around Column 1

Around column 2, and including F_{18p} and T_1 from the first reactor, there are six control objectives affected by the remaining seven manipulated variables (this is the third group in the elimination table). There are 13 combinations possible within this group and these are listed, with similar comments, in Table 3.

Control Objectives

		<u>C_{16b}</u>	<u>P_{c2}</u>	<u>H_{rd2}</u>	<u>H_{sp2}</u>	<u>T₁</u>	<u>F_{18^p}</u>	<u>Comments</u>
Manipulated	1	L ₂	F ₆	F ₁₆	F ₁₉	F _{w1}	F _b	Direct
Variables	2	L ₂	F ₆	F _{w5}	F ₁₉	F ₁₆	F _b	cooling rate - level
	3	L ₂	F _{w5}	F ₁₆	F ₆	F ₁₉	F _b	V/F
	4	F ₆	F _{w5}	F ₁₆	L ₂	F ₁₉	F _b	reflux - base level
	5	F ₆	F _{w5}	L ₂	F ₁₉	F ₁₆	F _b	V/F
	6	L ₂	F ₆	F _{w5}	F ₁₉	F _{w1}	F _b	cooling rate - level
	7	F ₆	F _{w5}	L ₂	F ₁₉	F _{w1}	F _b	V/F
	8	L ₂	F _{w5}	F ₁₆	F ₆	F _{w1}	F _b	V/F
	9	F ₆	F _{w5}	F ₁₆	L ₂	F _{w1}	F _b	reflux - base level
	10	F ₆	F _{w5}	F ₁₆	F ₁₉	F _{w1}	F _b	Direct
	11	L ₂	F _{w5}	F ₁₆	F ₁₉	F _{w1}	F _b	Direct
	12	L ₂	F ₆	F _{w5}	F ₁₉	F _{w1}	F ₁₆	cooling rate - level
	13	F ₆	F _{w5}	L ₂	F ₁₉	F _{w1}	F ₁₆	V/F

**Table 3 Alternative Control arrangements around Column 2/
Reactor 1**

The total number of distinct regulatory combinations for the process is therefore $7 \times 13 = 91$.

The different distillation control schemes can be identified with previously discussed options for single composition control (Chpt. 2). Many of the schemes achieve composition control by varying the V/F ratio without changing product flowrates. Large changes in internal flowrates must be made to achieve small changes in product composition. The schemes labeled direct composition control are much more sensitive however because they alter the material balance across the column.

Some of the schemes include unfavorable loops. For example, scheme 4 for Column 1 uses reflux to control column level. The long time required for changes in liquid flow at the top of the column to reach the base mean that this loop would not normally be used. Some schemes have reflux drum level controlled by condenser cooling rate (scheme 1 for Column 1). As this introduces the non-linearity and lags associated with the condenser into a level loop it is difficult to see a situation where this would be favoured over using distillate or reflux flow.

In the options for temperature control in Reactor 1 there is the possibility of varying either the F_{19} or F_{16} flowrate through Exchanger 3 (schemes 2 - 5 around Column 2). Changes in F_{16} affect both the temperature and rate of this stream which will subsequently, on mixing with the make-up stream of pure B, alter the enthalpy of F_{18} , one of the two reactor feeds. This would be ineffective because of the relative size of the streams involved. F_{16} makes up only 16% of the reactor feed and consideration of the steady state disturbances that affect the reactor suggests that the F_{16} flow needs to be zero to maintain the required feed preheat to keep the reaction temperature constant if the expected turndown in flowrate of pure A occurs. This kind of fluctuation in product flow would make control and operation of Column 2 extremely difficult. Variation of F_{19} flowrate would alter the temperature of F_{16} within a narrow band which would again be insufficient to handle the size of the expected disturbances. This option would also introduce the lags associated with Heat Exchanger 3 into the sensitive exothermic reactor temperature control loop. These are not therefore practicable alternatives for reactor temperature control

Another option is to replace variation in the rate of F_b with variation in F_{16} flow to control the F_{18} flowrate (schemes 12 and 13 around Column 2/Reactor 1). The same argument about disturbance size applies here. The flowrate of the make-up stream of pure B must change from 14000 Kg/hr to 11512 to maintain the ratio with the A flowrate when it turns down by 950 kgs/hr. This would become a reduction in F_{16} flowrate of from 3816 Kg/hr to 1328 Kg/hr which would completely disrupt the mass balance over Column 2.

Many of the configurations are therefore ruled out by real control requirements but there is no way of extracting this information from the controllability studies. However coupling the elimination information with an expert system provides guidance on useful control configurations and therefore which variables should be left out.

A further problem with the solution suggested by Johnston et al is that the COORDINATOR doesn't have non-zero entries in the matrix to show that distillate could be used to control composition (these entries appear at F_{11} - C_{11p} and F_{16} - C_{16b} in the matrix shown in Fig. 5). In the equations describing the column behaviour used to develop the COORDINATOR the distillate rate affects only a single state, the reflux drum level. Changes in reflux drum level do not directly affect any other system state, including the top composition. Therefore, a "direct access path" does not exist between distillate and composition and there can be no entry in that position in the matrix. The loop relies on reflux drum level control by reflux flowrate to transmit changes in distillate flowrate back to the column. This is a popular control alternative

recommended by Shinskey (1977) and McCune and Gallier (1973). If the two additional entries are added to the matrix and all the possible output sets found there are now 240 distinct possibilities. There are an extra 5 configurations around Column 1, listed in Table 4.

		Control Objectives				
		P _{c1}	C _{11p}	H _{rd1}	H _{sp1}	Comments
Manipulated	1	F _{s2}	F ₁₁	F _{w4}	L ₁	poor level loops
Variables	2	F _{s2}	F ₁₁	F _{w4}	F ₁₂	cooling rate - level
	3	F _{s2}	F ₁₁	L ₁	F ₁₂	Indirect composition
	4	F _{w4}	F ₁₁	L ₁	F _{s2}	mass balance problems
	5	F _{w4}	F ₁₁	L ₁	F ₁₂	Indirect composition

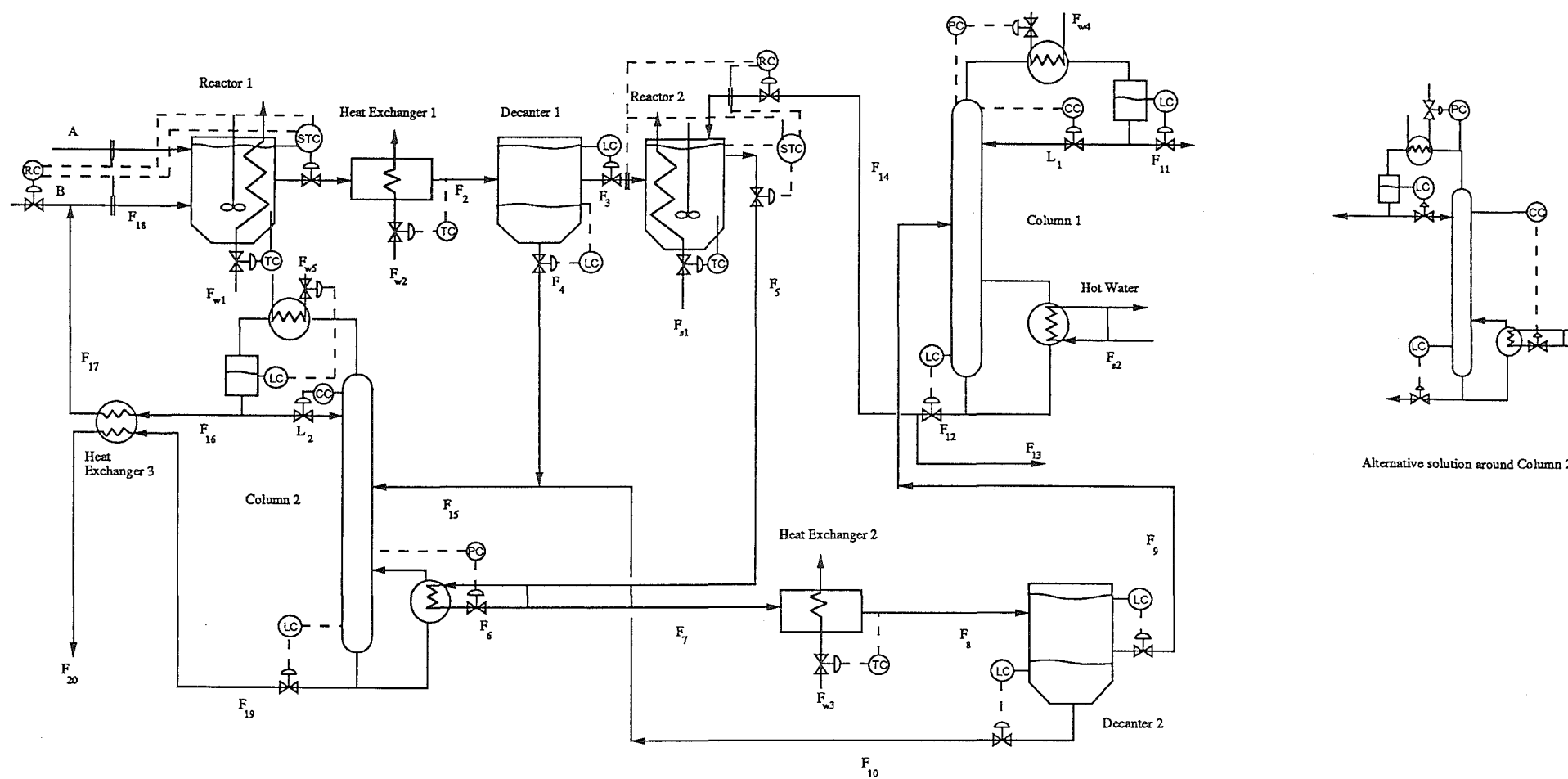
Table 4 Extra control configurations possible around Column 1

Some of the schemes have poorly functioning level loops (schemes 1 and 2) and scheme 4 cannot maintain a mass balance over the column because neither of the product streams is on level control. There are a further 7 possibilities introduced within the Column 2/Reactor 1 group, listed in Table 5

		Control Objectives						
		<u>C_{16b}</u>	<u>P_{c2}</u>	<u>H_{rd2}</u>	<u>H_{sp2}</u>	<u>T₁</u>	<u>F_{18p}</u>	<u>Comments</u>
Manipulated	1	F ₁₆	F ₆	L ₂	F ₁₉	F _{w1}	F _b	Indirect quality
Variables	2	F ₁₆	F ₆	F _{w5}	L ₂	F ₁₉	F _b	poor level loops
	3	F ₁₆	F ₆	F _{w5}	L ₂	F _{w1}	F _b	poor level loops
	4	F ₁₆	F ₆	F _{w5}	F ₁₉	F _{w1}	F _b	cooling rate - level
	5	F ₁₆	F _{w5}	L ₂	F ₆	F ₁₉	F _b	mass balance fails
	6	F ₁₆	F _{w5}	L ₂	F ₆	F _{w1}	F _b	mass balance fails
	7	F ₁₆	F _{w5}	L ₂	F ₁₉	F _{w1}	F _b	Indirect quality

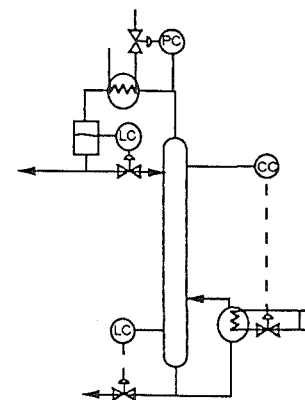
Table 5 Extra control pairing possibilities around Column 2/Reactor 1

The total number of possible control systems for the process is now $20 \times 12 = 240$ which adds further weight to the need for screening of the alternatives. The elimination data for the process is not changed by these modifications to the COORDINATOR.



STC = Spacetime controller

Fig. 6 Original solution using structural controllability with the alternative around Column 2



Alternative solution around Column 2

In many of the distillation control alternatives the remaining discarded manipulated variable should be placed on flow control. If reflux is excluded for instance it would be a constant flow decided by the hydraulic driving force between reflux drum and column. This would make feedrate changes to the column difficult as there would be no way of altering reflux flow to handle increased rates. The same argument applies to heat input, bottoms and distillate flows. Placing them on flow control does not change the controllability analysis because the extra control objective introduced is paired with itself. Uncontrolled streams of this kind, regarded as disturbances, are allowed by controllability but not in practice.

4.2 Problem Solution Using Expert Systems

The problem was solved using the program described and the alternative solution is shown in Fig. 7. The program was given almost the same calculation order as in the original paper except Reactor 1 was not coupled with Heat Exchanger 3 and a mixing tee was added to handle the mixing of F_b and F_{17} to form F_{18} ie.

- 1) Column 1
- 2) Decanter 2
- 3) Heat Exchanger 2
- 4) Column 2
- 5) Reactor 2
- 6) Decanter 1
- 7) Heat Exchanger 1
- 8) Reactor 1
- 9) Mixing Tee (F_b and F_{17})

The recommended control systems for each of the unit operations and a comparison with the Johnston et al solution follows.

4.2.1 Column 1

The program began by suggesting 8 possible schemes, shown in appendix AI, for Column 1. These correspond to schemes 2, 3, 5, 6 and 7 in Table 2 and schemes 3, 4 and 5 in Table 4. Extra schemes allowed by structural controllability that include unworkable loops such as reflux with column level and condenser cooling rate with reflux drum level are screened out by the program. There are comments on each of the suggested schemes identifying which control composition directly or indirectly

or by using the V/F ratio and whether any have mass balance difficulties. These comments are useful to the designer in making his final decision on the basic regulatory structure. Johnston et al chose a direct composition control scheme (scheme 7 in Table 2). A better choice would have been the indirect composition control scheme shown in fig.7 (scheme 5 in Table 4) because this approach handles best the upsets in reflux temperature, caused by changes in condenser duty. The reflux flow - reflux drum level loop compensates internal reflux for these changes. This scheme could also be improved by a further addition more easily than the Johnston solution. The feed forward scheme shown in Chpt. 2, Fig. 12 could be added to handle the expected changes in feed rate caused by changing the feedrate of A to the process. The only real advantage that the Johnston solution has is faster dynamics in the composition loop. However, the lag introduced by the reflux drum in the indirect scheme can be removed using the feed forward modification of Chpt. 2, Fig. 5. The V/F options can probably be discounted although the argument about the relative sensitivity of the two classes of composition control should be checked by steady state calculations.

After making the final choice the user enters the manipulated variables paired with their respective control objectives and the program forms a list of them, in this case $[F_{w4}, F_{11}, L_1, F_{12}]$. This is compared with the original set of manipulated variables available to the column $[F_9, F_{11}, F_{12}, L_1, F_{s2}, F_{w4}]$ to establish that F_9 and F_{s2} have been left out of the local control system. The program detects that F_9 can be used to control level in the second decanter so it has not been entirely left out of the plant control system. F_{s2} can not be used anywhere else however so it is added to the eliminated variables list. The second group in the elimination information in fig. 5 shows that it is possible to leave out one of the group $[F_{s2}, F_{w4}, L_1, F_{11}, F_{12}]$ so the plant remains controllable after this choice of control system for Column 1.

4.2.2 Decanters 1 and 2

The pairings for these two unit operations are forced which is recognized and reported by the program. This is, of course, the same as the solution by Johnston et al. There is no further advice available on control of this type of unit operation although an expert module could be added to the program in the future.

4.2.3 Heat Exchangers 1 and 2

Once again the basic regulatory pairing is forced and is the same in both solutions, T_8 must be paired with F_{w3} and T_2 with F_{w2} . However the expert system recommends feed forward control to cope with the expected changes in product flow from Reactors 1 and 2. This would improve control of the stream temperature into the

decanters and consequently prevent upsets in the separation of G from the light phase. This is an improvement that can't be part of a solution based only on controllability.

4.2.4 Column 2

The same possibilities and comments are reported by the program for Column 2 as appeared for Column 1. Similar arguments for choosing the indirect composition control scheme as were used in that case can be put forward here. Both of the solutions suggested by Johnston et al are V/F control alternatives (schemes 6 and 7 in Table 3) which are unlikely to provide good composition control and one includes the unacceptable condenser cooling rate - reflux drum level loop.

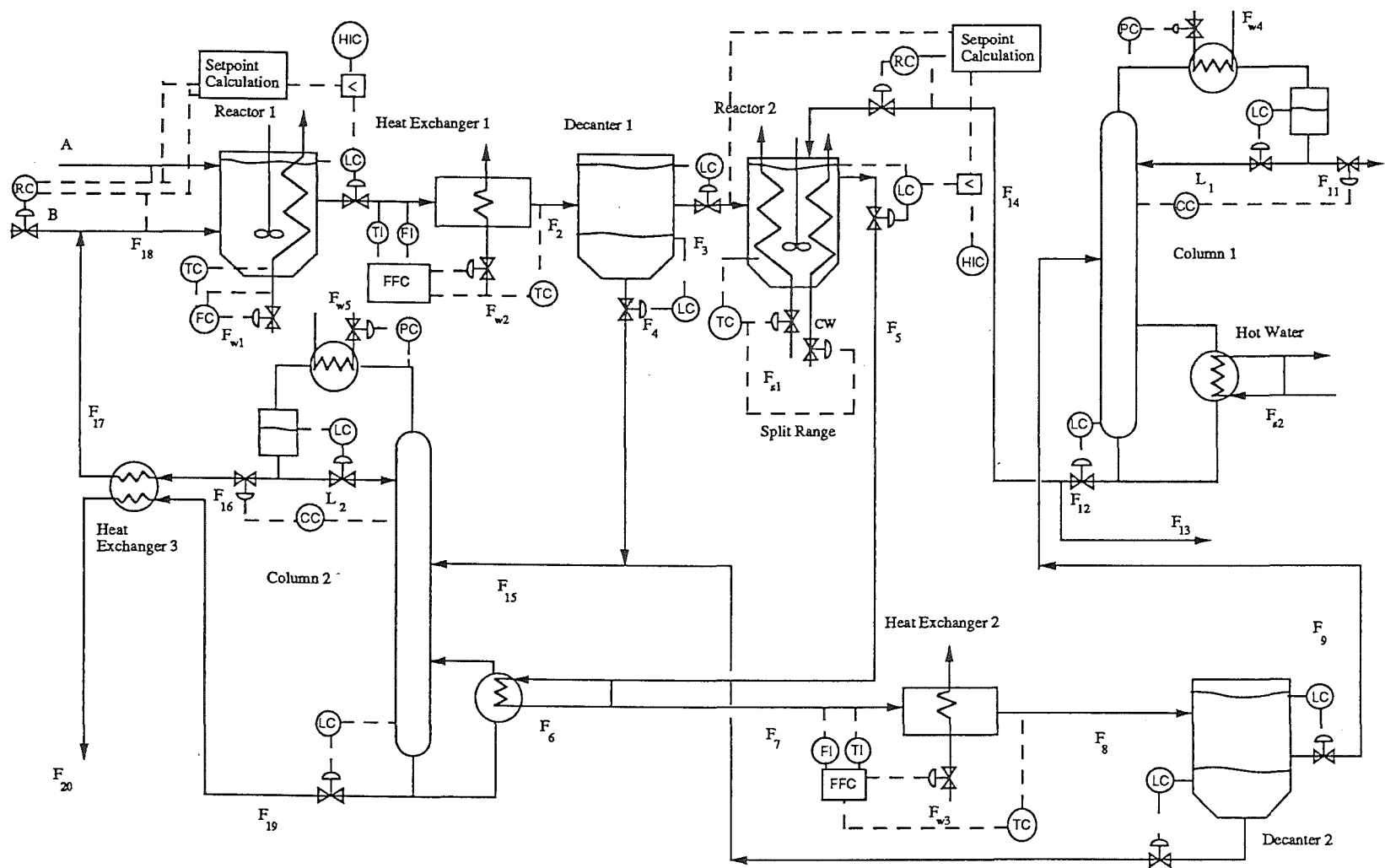
4.2.5 Reactor 2

The second reactor is an unusual case which was not considered in the original rule base for reactor control. Although it is an exothermic reactor the feed preheating to the reaction temperature significantly exceeds the heat released by the reaction. Therefore the reactor must be heated. However, it is possible that situations may arise where cooling is required. For example, if the feed is reduced significantly or shutdown completely the quenching effect would be removed and a reaction runaway might occur if cooling were not available. A safer system would be to have a split range temperature controller manipulating heat input or cooling rate as required. An extra rule was added to the database to cope with this situation.

The suggested method for space-time control on the reactors could lead to them overfilling as the level changes to compensate for throughput variations. There is a need for tighter specific level control to prevent this happening and the expert system rule reflects this. Inventory and composition control are achieved by placing the product on level control and controlling the ratio of the available feedrate to the second measured feedrate. A calculation block calculates the level controller setpoint from the feed flow measurements and a remote setpoint overrides this signal if the level gets to high. This necessitated an extra rule in the expert system because the original reactor module handled reactors where both feeds could be placed on flow control. The basic regulatory pairing for this system is forced and this is reported by the program.

4.2.6 Reactor 1

The inventory and composition control scheme for Reactor 1 is the same as described for Reactor 2 but this one is truly exothermic. The reactor is assumed stable so throttling the flow to the coil for temperature control is a viable option. The expert system also recommends a cascade improvement to this loop to cope with disturbances



FFC = Feedforward controller

Fig. 7 Alternative solution provided by an expert systems approach

in cooling water flow. Varying the feed temperature is another suggested possibility but as this can only be achieved by throttling F_{16} or F_{19} it is discounted. The more direct option of temperature control using F_{w1} was selected by the designer, as it was by Johnston. The other pairings around the reactor are forced and are reported by the program.

4.2.7 Mixing Tee

A mixing tee is considered as an unknown unit operation and because F_{16} is part of the control scheme for Column 2 the only alternative here is F_b . This is reported by the program and entered by the designer and the control system for the entire plant is complete.

5.0 Conclusion

The discussion in the previous two chapters and the example demonstrate the potential of expert systems to develop the basic regulatory structures derived from controllability analysis into a real solution for control systems synthesis for whole processes. The expert systems can provide guidance on which variables should be left out of the process control system by screening out unfavorable options, such as a distillation system with column level controlled by reflux rate and by passing information on about where the suggested ones are appropriate. Often the basic regulatory structure is predetermined and identified by controllability analysis. In these cases improvements to the regulatory system, such as feed forward control around a heat exchanger, are recommended should the situation require it. Constraint conditions are also recognised and control methods suggested to deal with them. The reactor level control and temperature control in the second reactor are examples of this. A drawback of the rulebased approach was also exposed by the example. If a situation was not originally envisaged by the author of the program no solution is forthcoming. The reactor module was unable to come up with an adequate control system for an exothermic reactor that required heating rather than cooling before an extra rule was added to the rulebase. As a program of this kind is used to tackle more examples the rulebase becomes more comprehensive. Unlike much conventional computer programming building an effective expert system is an iterative process.

The required procedures for matrix reordering and column elimination information preparation are written in Fortran while the coordinated knowledge based systems are implemented in Prolog. The integration of these two program sets remains as work for the future.

6.0 Nomenclature

F_A/F_{18} = The ratio of the two feedrates to Reactor 1

STR1 = The space-time in Reactor 1

T_2 = The temperature of the process stream leaving Heat Exchanger 1

V_{G1} = The volume of the heavy layer in Decanter 1

V_{E1} = The volume of the light layer in Decanter 1

F_3/F_{14} = The ratio of the two feedrates to Reactor 2

STR2 = The space-time in Reactor 2

T_8 = The temperature of the process stream leaving Heat Exchanger 2

V_{G2} = The volume of the heavy layer in Decanter 2

V_{E2} = The volume of the light layer in Decanter 2

T_5 = The operating temperature of Reactor 2

F_{18}^P = The flowrate of stream 18 (as a control objective rather than manipulated variable)

P_{c1} = The operating pressure of Column 1

C_{11P} = The concentration of P in stream F_{11}

H_{rd1} = The liquid holdup in the Column 1 reflux drum

H_{sp1} = The liquid holdup at the base of Column 1

C_{16B} = The concentration of B in stream F_{16}

P_{c2} = The operating pressure of Column 2

H_{rd2} = The liquid holdup in the Column 2 reflux drum

H_{sp2} = The liquid holdup at the base of Column 2

T_1 = The operating temperature of Reactor 1

Chapter 7 - Conclusions and Recommendations For Future Work

As part of the continuing research in Computer Aided Design to develop a software tool for the synthesis of control systems for chemical plants this project set out to explore the prospects for a solution using expert systems. Knowledge acquisition lead to prototype programs for the more common unit operations; distillation columns, heat exchangers and reactors. These were coordinated into an integrated system for control system synthesis for whole plants using concepts from previous research into the structural controllability of processes. During the research work some strengths, weaknesses and opportunities for the future were revealed.

1.0 Structural Controllability

The current research project has revealed areas where the techniques taken from structural controllability could be improved or expanded. The algorithm developed by Johnston and Barton for finding the elimination information for a structural matrix could be made more computationally efficient by modifying it into a form similar to Duff's algorithm as suggested in Appendix AII. The differences between the elimination information from the normal approach and that from the formation of augmenting paths demonstrated in the appendix need to be resolved before this would be possible.

The elimination information itself provides a basis for the decomposition of a process into smaller pieces for the application of modern control techniques. Conventional wisdom would see this decomposition made along unit operations boundaries. However if the example of the Williams-Otto plant from Chapter 6 is used the manipulated variables and control objectives fall into 3 separate groups. The first group (refer to Chapter 6, Fig 5) is manipulated variables and control objectives that can be paired in only one way. This requirement means that a Multi-Input Multi-Output (MIMO) controller wouldn't be appropriate within the group and would not improve control performance over conventional SISO techniques. The second group in the Table could form the basis for a non-square MIMO controller with 5 inputs and 4 outputs which would operate completely separately from a further MIMO controller

with 7 inputs and 6 outputs corresponding to the third group in the table. The two groups are unconnected because the manipulated variables in one group don't affect the control objectives in the other as shown in Fig. 1 below. This is an unexpected spinoff of applying the elimination algorithm to the COORDINATOR and should be developed further on other examples too see if it is a generally useful concept.

	F _{s2}	F _{w4}	L ₁	F ₁₁	F ₁₂	F _b	F ₁₆	L ₂	F ₆	F _{w5}	F ₁₉	F _{w1}
P _{c1}	x	x										
C _{11p}	x		x	x								
H _{rd1}		x	x	x								
H _{sp1}	x		x		x							
F _{18P}						x	x					
C _{16b}							x	x	x			
P _{c2}										x	x	
H _{rd2}							x	x			x	
H _{sp2}								x	x			x
T ₁							x				x	x

Fig. 1 Matrix showing 2 unconnected groupings of variables

The equations which form the initial structural models for unit operations need to contain all the information for control systems synthesis otherwise some options are missed. For example, if the connection between distillate or bottom product flowrate and composition is left out then useful control alternatives for distillation column's are not considered.

2.0 Individual Unit Operation Expert Systems

There are a number of conclusions that can be drawn and improvements that can be made to the individual unit operation expert system modules.

2.1 Distillation Expert System

The current distillation control expert system is able to handle the formulation of the basic regulatory control system for two product columns effectively. The combinatorial difficulties associated with this aspect of the problem are dealt with by

screening out all but the more attractive options, a process that is done cheaply using heuristic knowledge and inexpensive calculations (the Relative Gain Array). There are some short term improvements already possible at this stage in the knowledge acquisition process. Once the decision on the final regulatory structure is made by the user, aided by the program, a further rulebase should be added to make recommendations on additions that would improve control quality. This would include recommending feed forward control to handle feedrate changes, additions to indirect composition control schemes to remove accumulator lag from the composition control loop and internal reflux control and tray pressure drop control to prevent flooding or weeping.

If further knowledge acquisition work is done then the total number of rules could be increased to handle more complicated columns including those with a sidestream product, columns designed for light ends and heavy ends removal from feedstock and multicomponent columns that require dual composition control. The rulebase is, at the moment, heavily influenced by the work of Shinskey especially in considering dual composition control. There are at least two other recent distillation control texts (Buckley et al. 1985 and Deshpande, 1985) that should be reviewed more extensively than was possible in the time available as part of the continuing knowledge acquisition process and the knowledge base broadened to include their expertise.

Distillation column control system synthesis has attracted the most interest as a target for the application of expert systems. The published papers so far however lack detail about the composition of the rulebase possibly because the information is proprietary or there is insufficient room in the articles. The rules identified in this work are all available in the appendices to the thesis. This work differs from others in several areas. The structure of the program, arising as it does from the use of a Prolog development tool which is primarily a backward-chaining inference engine, is similar to that of Birky and McAvoy but differs from the work of Shinskey and Niida. The depth of organisation of knowledge about pressure control is not apparent in any of the other work and the incorporation of Relative Gain Array calculations along with heuristics to determine dual composition control configurations is also new. The philosophy of the program in proposing a number of solutions with comments and allowing the final decision to rest with the designer is also different as the others seem to produce only a single solution. This is not a desirable result for a program that relies only on heuristics

The program recommends a number of possibilities with further comments about their usefulness. This could cause confusion for an inexperienced user especially if competing circumstances apply. In distillation, for example, if the top product feeds

another column this would favour a direct composition control scheme but if there were upsets expected in the overhead cooling as well that would favour an indirect composition control scheme. These conflicts would be resolved subjectively by an expert. A ranking system that modeled this thought process would be a useful addition to the program. In a recent paper by Fan et al (1987), on the sequencing of distillation columns, a ranking system for alternative separation sequences was developed based on fuzzy logic. A similar approach may be possible in the control systems synthesis area.

2.2 Heat Exchanger Expert System

There is very little published material on control systems synthesis for heat exchangers using expert systems. The material in this thesis therefore provides a good basis for building up the necessary heuristics for a useful program.

The expert system can successfully handle the recommendation of common control systems for heat exchangers. This requires the selection of the appropriate configurations or improvements (feed forward or cascade control) based on expected disturbance size rather than the extensive variable pairing problem required for columns. The program would be improved by adding a more numerical dimension to its decisions. The non-linearity of response of a heat exchanger ultimately decides the size of steady state disturbance that a particular heat exchanger will successfully handle. The program should at least be able to make simple steady-state calculations to check out the rangeability of control possible for a heat exchanger and suggest whether the configurations are really able to cope with the known disturbances.

The program brings in the extra dimension from outside control science of the effect of bypass valve selection and condensate removal method on control. These points are vital for adequate control but are not considered by any theoretical techniques.

2.3 Reactor Expert System

As there is little published on this subject either the same comments that were made about this work in relation to heat exchanger expert systems can also be made about reactors. Even at this stage in its development the system can handle synthesis of control for a wide variety of continuous stirred tank, tubular and fixed bed reactors but it would benefit from a broadening of the rule base. In a number of examples, situations arose that were not originally envisaged by the author such as an exothermic

reactor with a feed preheat requirement that exceeds the reaction heat release. The rulebase could be added to and become more comprehensive as it was tried on more examples. The same comment also applies to the other unit operation advisory systems. The stability of exothermic reactions is an important aspect of reactor control. In the current expert system this is assessed by heuristics. The program should however be able to calculate numerical criteria, of the type described in Chapter 4, and warn the user of any stability problems in the design.

Many of the reactor configurations have an associated control type. For example, a jacket cooled reactor with a once through flow of coolant typically has reaction temperature controlled by varying coolant flow. A frame based system of knowledge representation would be useful here as the typical control system might fill a "slot" in the frame describing this class of reactor. After a reactor is specified the program would "know" immediately the likely control possibilities from the appropriate values inherited from the parent frame for this reactor and could subsequently check stability and control quality aspects for each without prompting from the user. A similar knowledge representation strategy would be possible for the other unit operations as well and may have considerable advantages over the purely rule-and-fact approach used up to now. Frames have become a popular knowledge representation technique in work in this area (Niida and Umeda, 1986, Birky and McAvoy, 1988 and Stephanopoulos et al, 1987). The third reference especially draws extensively on this idea as the work is carried out using the KEE system from Intellicorp which uses the frame as its basic knowledge representation technique.

3.0 The Whole Plant Control Systems Synthesis Program

The present work suggests an approach to control systems synthesis based on expert systems for the unit operations that make up the process coordinated by ideas from the structural controllability of whole processes. This is a successful approach and clearly different from previous research. It is difficult to gain more than general concepts from what has been published up to now whereas this work is much more detailed.

The present suggested approach to whole plant control system synthesis is implemented in a number of unconnected programs. The structural controllability aspects of the work are predominantly written in Fortran, although the elimination algorithm was translated into Prolog, and are separate from the synthesis program written in Prolog (the coordinated expert systems). The programs should be integrated into a complete package that allows the user to approach all aspects of the problem as

required. The user should be able to enter details of the COORDINATOR matrix and have the program reorder the matrix into a block form, pinpoint control objectives that have to be eliminated from consideration for controllability and calculate the elimination information for the matrix. The programs should communicate through a shared database so this data and the forced pairing information should be passed to the integrated synthesis program described in Chapter 6 and the synthesis completed along the lines described in the thesis. Advice on the control of other unit operations apart from heat exchangers, distillation columns and reactors should be included in the future to broaden the applicability of the program.

Beyond these more obvious ideas is a significantly expanded program which carries out control system synthesis using the approach summarised in Chapter 1, Fig. 9. This program would be able to call upon analytical techniques such as condition number calculation for candidate control schemes, steady-state arguments (examples of these were used in Chapter 6 to rule out control systems which couldn't cope with the expected disturbances), stability analysis and even dynamic simulation where appropriate to select an optimum control system. It could be extended to include a program that would select control objectives for a specified process using heuristics and analytical techniques, some of which are described in the introductory chapter (Morari et al, 1980, Fisher et al, 1988a) and would therefore cover the whole scope of the problem of control system synthesis for chemical processes.

4.0 Impact of Problems and Improvements in Expert Systems Technology

It is generally recognised that rule based expert systems have a number of associated difficulties that mean at times they will perform in a way that is far from "intelligent". They are inflexible when presented with a situation not originally thought of by the author of the system and their performance degrades rapidly when there are no rules in the rulebase to handle a problem. There are instances of this in the current work described elsewhere in the thesis. More advanced expert system techniques such as qualitative simulation and causal reasoning may evolve into a knowledge representation structure that overcomes these problems by exposing more of the "deep" or fundamental knowledge leading to more robust programs. A sophisticated expert system would be able to reason from the feed conditions and expected heat release of a reaction whether heating or cooling was needed to control the reactor for example.

Debugging and testing a system involves checking every feasible path for accuracy, a tedious task that rapidly grows as the rulebase enlarges. This does not

present a problem in the present work because the rulebases are comparatively small but if they were expanded as suggested it would require considerable labour. There are successful expert systems running commercially with rulebases of several thousand rules however.

A strength of these types of programs is claimed to be an easy to understand and maintain rulebase. Often however rules are added which control the inference procedure rather than contributing anything to it. This sort of problem was encountered when writing the earliest distillation expert system using an expert system shell. Control rules had to be added before a sensible result could be obtained (Earl and Williamson, 1988). The Prolog versions are an improvement on this but in order to increase computational speed the code is cryptic and not easily understood by an uninformed user.

Explanation of reasoning is only partially answered by the device of having the reason for many of the user directed queries available on request and further descriptions of recommended control systems made after the program has run. Cogent explanation of reasoning is an important part of the educational and debugging aspects of an expert system. As control system synthesis is likely to be the design activity that a chemical engineer knows least about the educational and therefore explanation capabilities of a system are important. It could be helpful, for example, if the user has a control system that he would like to use that was different from what had been recommended if it were possible for the program to explain why this is less likely to be successful. This kind of "why not" capability is not part of contemporary technology in expert system shells, which handle "why" and "how" queries well, and has not been addressed in this work but should be in the future.

Handling uncertain information or even "don't know" answers is a significant part of expert system technology. This issue has not found a place in the present work although a possible application is suggested in the area of distillation column control to rank alternatives. The handling of "don't know" requests would be important as the system should be able to provide helpful advice even if some input information is unavailable. This is a built-in facility in some shells but is not yet included in this work.

All the programs written so far need improvement to the user interface before they could confidently be approached by the uninitiated user. The ideal method for reporting solutions for example, is by explanatory text and graphics. A PID showing recommended solutions in the customary pictorial language of the engineer is essential. Birky and McAvoy (1988) have made some progress towards this goal as has

Shinsky (1986). The extension of the current work in this direction is important if these programs are ever to find widespread acceptance among practicing engineers. In this area the input of experienced programmers to such a project would be useful.

The continuing identification of appropriate heuristics is also important and is far less advanced in control system synthesis than in process synthesis. An additional way of achieving this rather than using texts and articles on control would be to have an industrial expert as part of the team setting up such a system. Conventional expert system wisdom recommends having a domain expert and knowledge engineer, conversant with expert systems techniques, as the key members in the development team.

Shallow knowledge expert systems in their current form are only able to provide useful advice on problems envisaged by their author. This is already sufficient, if the rulebase is extensive enough, for relatively inexperienced engineers to use to solve real problems effectively. As the knowledge within the system becomes deeper and includes access to appropriate analytical techniques, coordinated expert systems should provide an even more useful solution to computer aided whole plant control systems synthesis problem.

References

Allen R. M., Seminar on Computer Control for the Process Industries, University of Canterbury, October, 1986.

Arkun Y., Chemical Process Control - CPC-III, p 323, 1986.

Barton G. W., Chan W-K, Perkins J. D. and Prince R. G. H., Proceedings of CHEMECA 86, pg 200, 1986.

Birky G. J., McAvoy T. J. and Modarres M., Comp. and Chem. Eng., 12, (9/10), pg 1045, 1988.

Bodman S. W., " The Industrial Practice of Chemical Process Engineering", M. I. T. Press, 1968.

Bristol E. H., Chem. Eng. Prog., 76, (11), pg 84, 1980.

Buckley P. S., "Techniques of Process Control", John Wiley & Sons, New York, 1964.

Buckley P. S., Luyben W. L. and Shunta J. P., " Design of Distillation Control Systems ", Instrument Sociey of America, 1985.

Chin T. G., Hydrocarbon Processing, pg 145, October, 1979.

Despande P. B., " Distillation Dynamics and Control ", Instrument Society of America, 1985.

Duff I. S., Trans. of Mathematical Software, 7, pg 315, 1981.

Earl W. B. and Williamson C. J., Proc. of the 3rd Int. Symp. on Process Systems Engineering, Sydney, Australia, pg 175, 1988.

Fan L. T., Gharpuray M. M. and Huang Y. W., Chem. Eng., p 57, March 2nd, 1987.

Fisher W. R., Doherty M. F. and Douglas J. M., Ind. Eng. Chem. Res., 27, p 597, 1988a.

Fisher W. R., Doherty M. F. and Douglas J. M., Ind. Eng. Chem. Res., p 606, 1988b.

Fisher W. R., Doherty M. F. and Douglas J. M., Ind. Eng. Chem. Res., 27, p 611, 1988c.

Forman E. R., " Control Dynamics in Heat Transfer", Chemical Engineering, Jan 3, 1966.

L. M. Gordon, "Handbook of Advanced Control Systems and Instrumentation ", pg 174, Gulf Publishing Co., 1987.

Govind R. and Powers G. J., AIChEJ, 28, (1), pg 60, 1982.

Harriott P., " Process Control ", McGraw-Hill, 1964.

Holt B. and Morari M., Chem. Eng. Sci., 40, (7), pg 1229, 1985.

Johnston R. D., PhD Thesis, University of Sydney, 1985.

Johnston R. D. and Barton G. W., Int. J. Control, 41, (6), pg 1477, 1985a

Johnston R. D., and Barton G. W., Proceedings of CHEMECA 85, pg 387, 1985b.

Johnston R. D., Barton G. W., and Brisk M. L., Int. J. Control, 40, (2), pg 257, 1984.

Johnston R. D., Barton G. W. and Brisk M. L., Comp. and Chem. Eng., 9, (6), pg 547, 1985a.

Johnston R. D., Barton G. W. and Brisk M. L., Comp. and Chem. Eng., 9, (6), pg 557, 1985b.

Kwakernaak and Sivan, "Linear Optimal Control Systems", Wiley, 1972.

Lau, Alvarez and Jensen, AIChE. J, 31, (3), pg 427, 1985.

Levien and Morari M., AIChE. J., 33, pg 83, 1987.

Lien K., Suzuki G. and Westerberg A. W., Chem. Eng. Science, 42, (5), pg 1049, 1987.

Liptak B. G., "Instrument Engineers Handbook", Revised Edition, Chilton Book Co., 1985.

Luyben W. L., "Control of Chemical Reactors", Module A5.5, AIChE Modular Instruction Series on Process Control, pg 34, 1981.

Mathur J., Chemical Engineering, pg 101, Sept. 3, 1973.

McCune and Gallier, ISA Transactions, 12, (3), pg 193, 1973.

Millen J. K., Nibble Mac, pg 11, July/August, 1987.

Morari M., Chem. Eng. Science, 38, (11), pg 1881, 1983.

Morari, M. "Studies in the Synthesis of Control Structures for Chemical Processes", PhD. Thesis, Univ of Minnesota, 1977.

Morari M. and Stephanopoulos G., AIChE. J, 26, (2), pg 232, (1980).

Morari M., Arkun Y. and Stephanopoulos G., AIChEJ, 26, (2), pg 220, 1980.

Niida K. and Umeda T., Chemical Process Control - CPC-III, pg 869, 1986.

Nishida, Stephanopoulos and Westerberg, AIChEJ, 27, (3), pg 321, 1981.

Perkins J. D. and Wong M. P. F., Chem. Eng. Res. Des, 63, pg 358, November, 1985.

Perry and Chilton, " Chemical Engineers Handbook ",5th Edition, 1973.

Prassinos A. G., McAvoy T. J. and Bristol E., Proceedings 9th IFAC World Congress, Vol 3, Budapest, Hungary, 1984.

Rademaker, Rijnsdorp and Maarleveld, " Dynamics and Control of Continuous Distillation Units ", Elsevier, 1975.

Russell L. W. and Perkins J. D., Chem. Eng. Res. Des., 65, (11), pg 453, 1987.

Russell L. W., Perkins J. D. and Pantelides C. C., "Structural Controllability of Plants Described by Mixed Systems of Differential and Algebraic Equations", unpublished, 1988.

Rowney J. E., Jeffery L. E., Lang A. J. and Hatchman P. V., Proceedings of CHEMECA 87, Vol 2, pg 75.1, 1987.

C. J. Ryskamp, Hydrocarbon Processing, pgs 51, June, 1980.

Sanders C. W., Chemical Engineering, pg 145, Sept 21, 1959.

Shinskey F. G., Chemical Process Control - CPC-III, p 895, 1986.

Shinskey F. G. , " Distillation Control ", 1st Edition, McGraw-Hill,1977.

Shinskey F. G. , "Distillation Control ", 2nd Edition, McGraw-Hill, 1984.

Shinskey F. G. , "Process Control Systems", 2nd Edition, McGraw-Hill, 1979.

Stephanopoulos G., " Chemical Process Control ", Prentice-Hall, 1984.

Stephanopoulos G., Comp. and Chem. Eng., 7, (4), pg 331, 1983.

Stephanopoulos G., Johnston J., Kriticos R., Lakshmanan R., Mavrouvuniotis and Siletti C., Comp. and Chem. Eng., 11, (6), pg 655, 1987.

Swaney R. E. and Grossmann I. E., AIChE. J., 31, pg 621, 1985.

Takamatsu T., Hashimoto I. and Hashimoto Y., Int. Chem. Engineering, 27, (4), pg 669, 1987.

Umeda T., Kuriyama T. and Ichikawa A., Proceedings of the 7th IFAC World Congress, Helsinki, pg 271, 1978.

Umeda T. and Niida K., Control Theory and Advanced Technology, 2, (3), pg 385, 1986.

K. V. Waller, D. H. Finnerman, P. M. Sandelin, K. E. Hagglblom and S. E. Gustafsson, Ind. Eng. Chem. Res., 27, pg 624, 1988

R. A. Weiss C. Cheng, R. Derry and R. Johnston, Technical Papers, "Advanced Process Control", Warren Centre for Advanced Engineering, University of Sydney, 1987.

Williams T. J. and Otto R. E., AIEE Transactions, 79, pg 458, 1960.

Appendix AI - Interaction With The Control System Synthesis Package

1.0 Nomenclature

qe = column top composition

a = reflux accumulator level

lb = column base level

p = pressure

h = heat input at the column reboiler

d = distillate flowrate

b = bottom product flowrate

r = reflux flowrate

The control objectives and manipulated variables for the example follow the same nomenclature as defined in Chapter 6.

2.0 Interaction between the User and the program

(The following is a complete transcript of the interaction between the user and the program for the modified Williams-Otto plant example described in Chapter 6. The program begins by asking a series of questions about the first distillation column which are answered by the user.)

Is the feed under accurate flow control [y/n] ? n

Is the feed flowing from a well mixed storage tank or does it have only a slowly varying composition [y/n] ? n

Is the feed quality (percent vapourized, subcooling) accurately controlled [y/n] ? n

Does the column have a [steady/zero/intermittent] vapour product ? zero

Is the column condenser air,cooling water or process cooled [air/water/process] ? water

Does the column operate at atmospheric pressure [y/n] n

Is the condenser partially flooded [why/y/n] n

Does the condensate enter the reflux drum below the liquid level [why/y/n] y

Is there a pressure equalising line between column and reflux drum [why/y/n] n

Is there a bypass between column and reflux drum [y/n] n

Is there a liquid distillate product [y/n] y

Does the distillate have too small a flow for reflux drum level control [y/n] n

Does the bottoms have too small a flow for column level control [y/n] n

Does the column have an overheads line < 150 mm diameter [why/y/n] n

Is the column top temperature < 50 degrees C [why/y/n] y

Does the tower remove a small lightends stream from the feed [y/n] n

Does the tower remove a small heavy ends stream from the feed [y/n] n

Does the tower have a sidestream product [y/n] n

(The program outputs the feasible alternatives for the described column to the printer)

DISTILLATION CONTROL ADVISORY SYSTEM. CASE Column 1

The column has a zero vapour product

The column condenser is water cooled

The condensate enters below the liquid level

Column top temperature is < 50 degrees C

LIST OF POSSIBLE CONTROL PAIRS

possible pair qe h

possible pair qe r

possible pair qe d

possible pair lb h

possible pair lb b

possible pair a d

possible pair a r

possible pair p cooling_water_flow

possible pair p h

POSSIBLE CONTROL SCHEMES

The control objectives are ["qe","a","lb","p"]

Case 1 manipulated variables are ["h","d","b","cooling_water_flow"]

This is a direct temperature or composition control scheme.

It offers fastest response to control action and is better than indirect schemes when product feeds another column.

Case 2 manipulated variables are ["h","r","b","cooling_water_flow"] representing v/f control. Control won't be as good as other temperature or composition schemes

Case 3 manipulated variables are ["r","d","h","cooling_water_flow"] representing v/f control. Control won't be as good as other temperature or composition schemes

Case 4 manipulated variables are ["r","d","b","cooling_water_flow"]

This is a direct temperature or composition control scheme.

It offers fastest response to control action and is better than indirect schemes when product feeds another column.

Case 5 manipulated variables are ["r","d","b","h"]

This is a direct temperature or composition control scheme.

It offers fastest response to control action and is better than indirect schemes when product feeds another column.

Case 6 manipulated variables are ["d","r","h","cooling_water_flow"] leading to massbalance problems. This scheme shouldn't normally be used

Case 7 manipulated variables are ["d","r","b","cooling_water_flow"]

This is an indirect temperature or composition control scheme.

It has a slower response than direct schemes but is better when bottoms or distillate is too small for level control.

Coupling reflux with reflux drum level gives best response when upsets affect overhead cooling (eg. fin/fan cooler).

Case 8 manipulated variables are ["d","r","b","h"]

This is an indirect temperature or composition control scheme.

It has a slower response than direct schemes but is better when bottoms or distillate is too small for level control.

Coupling reflux with reflux drum level gives best response when upsets affect overhead cooling (eg. fin/fan cooler).

(The user enters his choice of manipulated variables for the respective control objectives for the column aided by the above information from the program)

Pairing information for the

control objectives in unit operation column_1

Enter the manipulated variable paired with control objective c11p f11

Enter the manipulated variable paired with control objective pc1 fw4

Enter the manipulated variable paired with control objective hrd1 l1

Enter the manipulated variable paired with control objective hsp1 f12

(Decanter 2 is the next unit in the calculation order and there is no expert advisory system for it, all the pairings are forced however, which is reported by the program)

Control possibilities for dec_2

Pairing information for the

control objectives in unit operation dec_2

vg2 must be paired with f10

ve2 must be paired with f9

All the control objectives are paired in this unit operation

(The program questions the user on heat exchanger 2)

Enter the filename.extension for the file needed to

initiate the database for unit operation hx_2 heat.dat

This expert system recommends control systems for

1) Heat exchangers with no phase change

2) Steam heated exchangers

Enter the heat exchanger type [1/2] 1

HEAT EXCHANGER ADVISORY SYSTEM. CASE hx_2

Is the controlled stream heated or cooled [heat/cool] cool

Is the heating/cooling stream a process or utility stream [pro/ut] ut

Select expected disturbance characteristics with arrow key

The disturbance characteristics are fast and small

Is the major disturbance a flow variation [y/n/?] y

Select disturbed flow with arrow keys

The disturbed flow is the controlled stream

Is the utility stream cooling water [y/n/?] y

Is there an exchanger bypass on the controlled stream [y/n/?] n

Is there a bypass on the other stream [y/n/?] n

AI-4

Can the controlled stream be throttled [y/n/?] n

Will the cooling water exit temperature be below 50 degrees C if the flow is reduced below design [y/n/?] y

(The program makes control recommendations for this exchanger and also reports that the exit temperature must be paired with cooling water flow for control)

Recommended control types are ;

Use feed-forward control throttling the cooling water stream

Pairing information for the

control objectives in unit operation hx_2

t8 must be paired with fw3

All the control objectives are paired in this unit operation

(The program moves onto column 2)

Enter the filename.extension for the file needed to
initiate the database for unit operation column_2 test3

Is the feed under accurate flow control [y/n] ? n

Is the feed flowing from a well mixed storage tank or does it have only a slowly
varying composition [y/n] ? n

Is the feed quality (percent vapourized, subcooling) accurately controlled [y/n] ? n

Does the column have a [steady/zero/intermittent] vapour product ? zero

Is the column condenser air,cooling water or process cooled [air/water/process] ?
water

Does the column operate at atmospheric pressure [y/n] n

Is the condenser partially flooded [why/y/n] n

Does the condensate enter the reflux drum below the liquid level [why/y/n] y

Is there a pressure equalising line between column and reflux drum [why/y/n] n

Is there a bypass between column and reflux drum [y/n] n

Is there a liquid distillate product [y/n] y

Does the distillate have too small a flow for reflux drum level control [y/n] n

Does the bottoms have too small a flow for column level control [y/n] n

Does the column have an overheads line < 150 mm diameter [why/y/n] n

Is the column top temperature < 50 degrees C [why/y/n] y

Does the tower remove a small lightends stream from the feed [y/n] n

Does the tower remove a small heavy ends stream from the feed [y/n] n

Does the tower have a sidestream product [y/n] n

(The program suggests control alternatives for column 2)

DISTILLATION CONTROL ADVISORY SYSTEM. CASE Column 2

The column has a zero vapour product

The column condenser is water cooled

The condensate enters below the liquid level

Column top temperature is < 50 degrees C

LIST OF POSSIBLE CONTROL PAIRS

possible pair qe h

possible pair qe r

possible pair qe d
 possible pair lb h
 possible pair lb b
 possible pair a d
 possible pair a r
 possible pair p cooling_water_flow
 possible pair p h

POSSIBLE CONTROL SCHEMES

The control objectives are ["qe","a","lb","p"]

Case 1 manipulated variables are ["h","d","b","cooling_water_flow"]

This is a direct temperature or composition control scheme.

It offers fastest response to control action and is better than indirect schemes when product feeds another column.

Case 2 manipulated variables are ["h","r","b","cooling_water_flow"] representing v/f control. Control won't be as good as other temperature or composition schemes

Case 3 manipulated variables are ["r","d","h","cooling_water_flow"] representing v/f control. Control won't be as good as other temperature or composition schemes

Case 4 manipulated variables are ["r","d","b","cooling_water_flow"]

This is a direct temperature or composition control scheme.

It offers fastest response to control action and is better than indirect schemes when product feeds another column.

Case 5 manipulated variables are ["r","d","b","h"]

This is a direct temperature or composition control scheme.

It offers fastest response to control action and is better than indirect schemes when product feeds another column.

Case 6 manipulated variables are ["d","r","h","cooling_water_flow"] leading to massbalance problems. This scheme shouldn't normally be used

Case 7 manipulated variables are ["d","r","b","cooling_water_flow"]

This is an indirect temperature or composition control scheme.

It has a slower response than direct schemes but is better when bottoms or distillate is too small for level control.

Coupling reflux with reflux drum level gives best response when upsets affect overhead cooling (eg. fin/fan cooler).

Case 8 manipulated variables are ["d","r","b","h"]

This is an indirect temperature or composition control scheme.

It has a slower response than direct schemes but is better when bottoms or distillate is too small for level control.

Coupling reflux with reflux drum level gives best response when upsets affect overhead cooling (eg. fin/fan cooler).

(The user enters the selected manipulated variables)

AI-6

Pairing information for the

control objectives in unit operation column_2

Enter the manipulated variable paired with control objective c16p f16

Enter the manipulated variable paired with control objective pc2 fw5

Enter the manipulated variable paired with control objective hrd2 l2

Enter the manipulated variable paired with control objective hsp2 f19

(The program moves onto Reactor 2)

Reactor advisory system called for reactor reactor_2

Is the reactor;

1) A CSTR

2) A tubular or fixed bed reactor

Enter the type [1/2] 1

Enter the filename.extension for the file needed to
initiate the database for unit operation reactor_2 react.dat

Is the reaction

1) exothermic

2) endothermic

3) exothermic, feed preheating > reaction heat release

Enter the choice [1/2/3] 3

Enter the filename.extension for the file needed to
initiate the database for this cstr cstr.dat

Is the reactor;

1) A single-pass reactor

2) Part of a recycle loop

Enter the type [1/2] 1

Is there feeding the reactor;

1) A single liquid feed

2) Two liquid feeds

3) A gas feed and a liquid feed

Select the option that applies [1/2/3] 2

Is one or both feeds available for control [1/2] 1

Is conversion a control objective [y/n/?] y

Are variations expected in reactant feed composition [y/n/?] n

(The program recommends control strategies for temperature, inventory and composition for Reactor 2)

Recommended control types are;

Normally temperature control would be by varying heating rate but if reaction heat release exceeds feed preheat in abnormal operation cooling is required. Use a split range temperature controller manipulating heating rate or cooling when needed

Recommended inventory control types are;

The product should be on level control

Recommended composition control types are;

Place the available feed on ratio control to the other feed.

Calculate changes in set point for a level controller on the product

to maintain spacetime constant. Override this signal with a low signal selector and a remote controller if the level exceeds a limit value.

(The program reports the pairing information for the reactor)

Pairing information for the
control objectives in unit operation reactor_2
vol2 must be paired with f5
f3/f14 must be paired with f14
t5 must be paired with fs1
All the control objectives are paired in this unit operation

(The pairing for the next unit operation, Decanter 1, are forced and this is reported by the program)

Control possibilities for dec_1
Pairing information for the
control objectives in unit operation dec_1
vg1 must be paired with f4
ve1 must be paired with f3
All the control objectives are paired in this unit operation

(The next unit operation is Heat Exchanger 1)

Enter the filename.extension for the file needed to
initiate the database for unit operation hx_1 heat.dat
This expert system recommends control systems for
1) Heat exchangers with no phase change
2) Steam heated exchangers
Enter the heat exchanger type [1/2] 1
HEAT EXCHANGER ADVISORY SYSTEM. CASE hx_1
Is the controlled stream heated or cooled [heat/cool] cool
Is the heating/cooling stream a process or utility stream [pro/ut] ut
Select expected disturbance characteristics with arrow key
The disturbance characteristics are fast and small
Is the major disturbance a flow variation [y/n/?] y
Select disturbed flow with arrow keys
The disturbed flow is the controlled stream
Is the utility stream cooling water [y/n/?] y
Is there an exchanger bypass on the controlled stream [y/n/?] n
Is there a bypass on the other stream [y/n/?] n
Can the controlled stream be throttled [y/n/?] n
Will the cooling water exit temperature be below 50 degrees C if the flow is reduced below design [y/n/?] y

(The program makes control suggestions and reports the required pairing)

Recommended control types are ;
Use feed-forward control throttling the cooling water stream
Pairing information for the

AI-8

control objectives in unit operation hx_1

t2 must be paired with fw2

All the control objectives are paired in this unit operation

(The program moves onto Reactor 1)

Reactor advisory system called for reactor reactor_1

Is the reactor;

1) A CSTR

2) A tubular or fixed bed reactor

Enter the type [1/2] 1

Enter the filename.extension for the file needed to

initiate the database for unit operation reactor_1 react1.dat

Is the reaction

1) exothermic

2) endothermic

3) exothermic, feed preheating > reaction heat release

Enter the choice [1/2/3] 1

Is the heat evolution/volume

1) small

2) medium

3) large

Enter the choice [1/2/3] 1

Does the CSTR have

1) A jacket

2) A coil

3) No attached heat exchange

As a means of cooling. Select the option that applies [1/2/3] 2

Are variations in coolant supply temp expected [y/n/?] y

Enter the filename.extension for the file needed to

initiate the database for this cstr cstr.dat

Is the reactor;

1) A single-pass reactor

2) Part of a recycle loop

Enter the type [1/2] 1

Is there feeding the reactor;

1) A single liquid feed

2) Two liquid feeds

3) A gas feed and a liquid feed

Select the option that applies [1/2/3] 2

Is one or both feeds available for control [1/2] 1

Is conversion a control objective [y/n/?] y

Are variations expected in reactant feed composition [y/n/?] n

(The program recommends control options and reports on the pairing for the reactor)

Recommended control types are;

Control reactor temperature by varying feed temperature

Control temperature by throttling the coolant flow in the cooling coil

Cascade reaction temperature onto a coolant temperature control loop

Recommended inventory control types are;

The product should be on level control

Recommended composition control types are;

Place the available feed on ratio control to the other feed.

Calculate changes in set point for a level controller on the product to maintain spacetime constant. Override this signal with a low signal selector and a remote controller if the level exceeds a limit value.

Pairing information for the

control objectives in unit operation reactor_1

vol1 must be paired with f1

fa/f18 must be paired with f18p

(There is one variable where a choice of manipulated variable is possible and the program asks the user to supply the required manipulated variable)

Enter the manipulated variable paired with control objective t1 fw1

(The mixing tee is an unknown unit operation ,without any expert advice. The program reports the control possibilities based on structural controllability and asks the user for his choice. At this point in the synthesis there is only one.)

Control possibilities for mix_1

Control Objectives f18p

fb

Pairing information for the

control objectives in unit operation mix_1

Enter the manipulated variable paired with control objective f18p fb

(In conclusion, the program reports the entire control system pairing selection)

Pair f18p with fb

Pair t1 with fw1

Pair hsp2 with f19

Pair hrd2 with l2

Pair pc2 with fw5

Pair c16p with f16

Pair hsp1 with f12

Pair hrd1 with l1

Pair pc1 with fw4

Pair c11p with f11

Pair fa/f18 with f18p

Pair vol1 with f1

Pair t2 with fw2

Pair vg1 with f4

Pair ve1 with f3

Pair f3/f14 with f14

Pair vol2 with f5

AI-10

Pair t8 with fw3

Pair vg2 with f10

Pair ve2 with f9

Pair t5 with fs1

Appendix All- Duff 's Algorithm and the Johnston and Barton Alternative

Duff's algorithm (1981) was developed as a computer efficient method for finding an output set for a matrix and it can be used in controllability analysis to detect whether a structural matrix has full generic rank or not. It was felt that a modified form of Duff's algorithm could be a more efficient method for finding the elimination information for a structural matrix than the approach used by Johnston and Barton. The heart of the algorithm is the formation of "augmenting paths" to increase the size of the assignment set for a matrix.

The formation of the assignment set begins by taking each row of a matrix in turn and assigning it a column which has a non-zero entry in the row. If the current row has no as yet unassigned columns an augmenting path must be traced through the matrix. A non-zero entry in the row is chosen and a path traced to the row that this column is assigned to. If there is an unassigned column here it becomes the new assignment for this row and the previously used column is now free for assignment to the original row. If the newly visited row has no unassigned entries then the length of the path is increased by choosing a non-zero column, proceeding to its assigned row and trying to augment the assignment at this point. This continues until a row is found which has an unassigned entry and a reassignment can be made back through the chain. The path is prevented from visiting the same column more than once. If the search reaches a row where all the entries are either assigned or already on the path it backtracks to the previous row and attempts to increase the path length again using an as yet untried column. If the search exhausts these alternatives and backtracks to the starting row the assignment set can't be further increased in size. The matrix is therefore structurally singular and has less than full generic rank.

Although the algorithm was written specifically to find assignment sets for a matrix it was felt that forming augmenting paths might produce the same matrix elimination information as the algorithm developed by Johnston and Barton described in Chapter 5, Fig. 7. Two examples are now shown, in the first forming augmenting paths produces the same elimination information and in the second there are significant differences.

The first example matrix shown in Fig. 1 is the transposed portion of the COORDINATOR in Chapter 6, Fig. 5 where some choice is possible in eliminating the columns from the original matrix (these have now become the rows in this example). The algorithm assigns columns to each of the rows. As there are two more rows than columns there will be two rows that can't be assigned. When these rows are reached an augmenting path will be unsuccessfully traced through the matrix. The rows this path visits form a group and if one member of this group were left out an assignment could be made. This is interpreted to mean that if one of the group of columns in the original matrix that correspond to this row group is left out an assignment is still possible in the original. This is the elimination information for this matrix as previously defined in the thesis.

	<u>F_{18p}</u>	<u>P_{c1}</u>	<u>C_{11p}</u>	<u>H_{rd1}</u>	<u>H_{sp1}</u>	<u>C_{16b}</u>	<u>P_{c2}</u>	<u>H_{rd2}</u>	<u>H_{sp2}</u>	<u>T₁</u>
F _b	x									
F ₁₆	x							x		x
F _{s2}		x	x		x					
F _{w4}		x		x						
L ₁			x	x	x					
F ₁₁				x						
F ₁₂					x					
L ₂						x		x	x	
F ₆						x	x		x	
F _{w5}							x	x		
F ₁₉									x	x
F _{w1}										x

Fig. 1 The First Matrix used for a trial of Duff's Algorithm

An assignment is made easily up to row F₁₂ and at this point an augmenting path must be traced. It visits rows F₁₂, F_{s2}, F_{w4}, L₁ and F₁₁ and the search terminates unsuccessfully. If one of these rows was excluded a successful assignment could be made. This is equivalent to saying that if one of these columns was excluded from the original matrix a full assignment could still be made. The assignment process continues until the final row F_{w1} is reached. Another augmenting path, visiting rows F_{w1}, F₁₆, F_b, F_{w5}, F₆, L₂ and F₁₉, is traced unsuccessfully. One of these columns could also be left out of the original matrix and still allow a full assignment set to be made. These

two groups are the same as those identified by the Johnston and Barton algorithm when it is applied to this part of the COORDINATOR matrix (Chapter 6, Fig. 5).

In a second example the produced elimination information is not the same. The matrix is shown in Fig. 2 and is the transpose of a matrix taken from Johnston and Barton (1984), pg. 259.

	1	2	3	4	5	6	7	8	9	10
1	x									x
2	x									
3		x								x
4		x								
5			x							
6			x	x				x		
7				x	x	x				x
8				x	x	x	x		x	
9					x	x				
10					x					
11						x	x		x	
12							x	x	x	x
13							x	x	x	x

Fig. 2 The Second Example used as a Trial for Duff's Algorithm

The elimination information produced by the two algorithms is shown in Table 1. The rows in the example matrix are the same as the columns in Johnston and Barton's original matrix so there should be a direct equivalence between the two sets, as there was in the previous example. It is possible however, to violate the elimination data produced by the augmenting path approach and still form a successful output set for the matrix . If the rows 1, 3, and 13 are excluded for example then the assignment, shown as the bold entries in Fig. 2, is possible even though two members of the first set in Table 1 have been left out.

Although forming augmenting paths may be more computationally efficient than the Johnston and Barton algorithm there is some doubt about whether it forms the correct elimination information and this should be investigated further.

Rows visited by the Augmenting Paths (original matrix)	Johnston and Barton's Data	
(1 from each group can be eliminated)	Columns	Elimination No.
1, 2, 3, 4	1, 2	1
1, 2, 5, 6, 7, 8, 9, 10, 11, 12	3, 4	1
1, 2, 5, 6, 7, 8, 9, 10, 11, 13	5, 6	1
	5,6,7,8,9,10,11,12,13	2
	1,2,3.....,13	3

**Table 1. Elimination Information for the second example matrix
by the two alternative methods**

Appendix All - Rule Base For the Distillation Expert System

The detailed knowledge contained in the distillation control synthesis expert system is explained here. In the "formcombinations" section a default list of control objectives is set up ie. (a,lb,p). These are accumulator level, bottoms level and pressure.

1.0 Input Section

The program checks for the following input conditions;

- i) Flow stable
- ii) Composition stable
- iii) Feed quality (enthalpy) stable
- iv) Vapour product- either zero, steady or intermittent
- v) Whether the vapour product is sent to a lower or compressed to a higher pressure (if there is a steady vapour product).
- vi) Type of condenser either air, water or process cooled

These queries are followed by a series requiring yes (y), no (n) or why answers,

- i) Is the column running at atmospheric pressure if there is no vapour product.
- ii) Condenser partially flooded (unless the column is atmospheric or there is a steady vapour product) then condensate ex the condenser is available as a manipulated variable-
- iii) Whether the reflux drum is flooded or excluded (subject to the condition that there is a partially flooded condenser and a zero vapour product)

- iv) Condenser located above the reflux drum and the pipework between them designed for gravity flow (unless there is a steady vapour product, or a flooded drum and subject to the condition that the system has a partially flooded condenser)
- v) Whether the position of condensate entry to the reflux drum is below the liquid level (unless the column is atmospheric, has a flooded drum or a steady vapour product)
- vi) Whether the condensate is subcooled (subject to the conditions that there is a zero vapour product, a partially flooded condenser and the column runs without a flooded drum)
- vii) Whether there is a source of inert gas available at a pressure above column pressure (only if there is an intermittent vapour product)
- viii) Whether there is a pressure equalising line between the column and the reflux drum (except if the column is atmospheric, operates with a flooded drum or has a steady vapour product)
- ix) Whether there is a bypass between column and reflux drum (unless the column is atmospheric, has a flooded drum or a steady vapour product).
- x) Whether there is a liquid distillate product.
- xi) Distillate too small to control accumulator level (unless there is no liquid distillate product or the drum runs flooded).
- xii) Bottoms too small to control column level
- xiii) Column overhead line < 150 mm diameter (unless the column is atmospheric, has a flooded drum or a steady vapour product) then the overhead vapour flow is a manipulated variable
- xiv) Top Temperature < 50 degrees C (subject to the condition that the column has a water cooled condenser and the column is not atmospheric)
- xv) If the condenser has an air-cooled condenser checks if it has air flow control louvers
- xvi) If the condenser is air cooled checks if it has adjustable pitch blades on the fans
- xvii) Whether the tower removes lightends
- xviii) Whether the tower removes heavy ends

xix) Whether the tower has a sidestream product from the tower

The final question relates to the sidestream.

i) Phase of the sidestream- vapour or liquid (as long as a sidestream product has been specified)

The rest of the input section establishes a list of controlled variables.

It asks whether the following are control objectives, composition in the top of the column, composition in the bottom of the column (If two compositions are specified then the following sequence is avoided because these must all be available as manipulations), Top temperature, Bottom temperature, Liquid distillate flow, Vapour distillate flow, Bottoms flow, Heat input, Reflux, Sidedraw rate. If the distillate is in two phases, vapour and liquid, then reflux temperature is added to the list of control objectives.

2.0 Rule Base For the Formloops section

- 1) If the column is atmospheric then pressure is not paired with a manipulated variable (default configuration is (p, atmospheric))
- 2) If there is a steady vapour product sent to a higher pressure then control pressure using a compressor recycle.
- 3) If there is a steady vapour product sent to a lower pressure which is not flow controlled then control pressure by the flow of the vapour product.
- 4) If the column is not an atmospheric column and does not have a steady vapour product and two composition control is not specified a possible pair is p with heat input
- 5) If the column does not run with a flooded drum has a water cooled condenser and a top temperature < 50 degrees C and does not have a distillate in two phases then a possible pair is p with cooling water flow
- 6) If the condenser is process stream cooled and does not have a distillate in two phases and the drum does not run flooded then a possible pair is p with process stream flow

- 7) If the column does not have a steady vapour product and the vapour flow is available as a manipulation then a possible pair is p with vapour flowrate (reflux vent open)
- 8) If the column has a zero vapour product and overhead vapour flow is available as a manipulation then a possible pair is pressure with vapour flow (reflux vent closed)
- 9) If the column does not have a steady vapour product and the condenser runs partially flooded and the column and reflux drum are at equal pressure and the condenser is above the drum with the pipework designed for gravity flow between condenser and drum and the condensate enters below the liquid level then a possible pair is pressure with vapour flow to the condenser
- 10) If the column does not have a steady vapour product and the condenser runs partially flooded and the column and reflux drum run at the same pressure and the condenser is above the reflux drum and the pipework between them is designed for gravity flow then a possible pair is pressure with condensate flow exit the condenser
- 11) If the column has a zero vapour product and does not require two composition control and the drum runs flooded then a possible pair is pressure with reflux flowrate
- 12) If the column condenser is air cooled with fans with adjustable pitch blades then a possible pair is pressure with blade pitch except when there is a two phase distillate.
- 13) If the column condenser is air cooled with adjustable louvers then a possible pair is pressure with louver position except when there is a two phase distillate.
- 14) If the column does not have a steady vapour product and the condenser is partially flooded and there is a bypass between the column and the reflux drum then a possible pair is pressure with condensate flow with the reflux drum on pressure control using the bypass
- 15) If the column has a zero vapour product and there is a bypass between the column and the drum and the condensate is subcooled and the condenser is partially flooded and the condenser is not above the reflux drum with gravity flow designed pipework and the condensate enters the drum below the liquid level then a possible pair is pressure with a hot gas bypass
- 16) If the column has an intermittent vapour product and inert gas is available at higher than column pressure then a possible pair is pressure with a split range control method using reflux drum vent and an inert gas bleed

17) If the reflux rate is not a control objective and the reflux drum does not run flooded and the reflux is a manipulated variable then accumulator level can be controlled by reflux

18) If the distillate is not a control objective and the reflux drum does not run flooded and a liquid distillate is available as a manipulated variable and the distillate is not too small to control accumulator level then accumulator level can be controlled by distillate flow

19) If the sidedraw rate is not a control objective and the reflux drum does not run flooded and the sidedraw is liquid then accumulator level can be controlled by sidedraw rate

20) If the reflux drum is flooded or excluded then remove accumulator level from the control objectives list.

21) If the drum does not run flooded and the distillate is a vapour and the condenser is water cooled with a top temperature $< 50^{\circ}\text{C}$ then a possible pair is accumulator level with cooling water flow.

22) If the drum does not run flooded and the distillate is a vapour and there is a process cooled condenser then a possible pair is accumulator level with process cooling flow.

23) If the drum does not run flooded and the distillate is a vapour and there is an air cooled condenser with louvers then a possible pair is accumulator level with louvers.

24) If the drum does not run flooded and the distillate is a vapour and there is an air cooled condenser with adjustable pitch fan blades then a possible pair is accumulator level with fan blade pitch.

25) If the the bottoms rate is not a control objective and the bottoms rate is a manipulated variable and the bottoms rate is not too small to control column level then column level can be controlled by bottoms rate

26) If the heat input is not a control objective and the heat input is a manipulated variable then column level can be controlled by heat input

27) If the sidedraw is a control objective then the sidedraw can be under flow control

28) If the reflux is a control objective and reflux is available as a manipulated variable then the reflux can be under flow control

- 29) If the liquid distillate is a control objective and the distillate is available as a manipulated variable then distillate can be under flow control.
- 30) If the vapour distillate is a control objective then it can be on flow control.
- 31) If the bottoms rate is a control objective and bottoms is available as a manipulated variable then bottoms can be under flow control
- 32) If heat input is a control objective and heat input is available as a manipulated variable then heat input can be under flow control
- 33) If top temperature is a control objective and liquid distillate is available as a manipulated variable then distillate can control top temperature
- 34) If top temperature is a control objective and reflux is available as a manipulated variable then reflux can control top temperature
- 35) If top temperature is a control objective and heat input is available as a manipulated variable then heat input can control top temperature
- 36) If top temperature is a control objective and vapour product is not a control objective and vapour product is sent to a low pressure then vapour product flow can control top temperature.
- 37) if top temperature is a control objective and there is a vapour product sent to a higher pressure then a possible pair is top temperature with compressor recycle.
- 38 -42) as 33-37 but replace top temperature with top quality
- 43) If bottom temperature is a control objective and bottoms is available as a manipulated variable then bottoms can control bottoms temperature
- 44) If bottom temperature is a control objective and heat input is available as a manipulated variable then heat input can control bottom temperature
- 45)-46) as 43 and 44 but with bottom quality replacing bottom temperature.
- 47) If the distillate is two phase and reflux temperature is a control objective and there is a water cooled condenser and the column top temperature $< 50^{\circ}\text{C}$ then cooling water rate can control reflux temperature.
- 48) If the distillate is two phase and reflux temperature is a control objective and there is a process cooled condenser then process cooling stream rate can control reflux temperature.

49) If the distillate is two phase and reflux temperature is a control objective and the condenser has air flow control louvers then the louvers can control reflux temperature.

50) If the distillate is two phase and reflux temperature is a control objective and the condenser has adjustable pitch fan blades then the fan blade pitch can control reflux temperature.

3.0 Controllability Checks

1) Controllability check for any problem not involving two composition control. Succeeds if any combination is possible between the control objectives and manipulations.

2) Control check for the case of two composition control and a normal reflux drum (not flooded or excluded) simplifies to a check of whether pressure has any control options.

3) Control check for two composition control for a flooded drum configuration (has different control objectives to 2 above)

4) If none of the prior three control checks are successful the program prints out a message to the user to this effect and aborts the run.

4.0 Createloops Section

For single or no composition control the program forms all feasible combinations using backtracking. The following rules are included to output comments after a selection has been made;

a) V/F control when the distillate is single phase and if the control scheme includes

-a/r and te/h or ts/h

-lb/h and te/r (r=reflux, te=top temperature, ts=bottom temperature, h=heat input).

The same comments also apply when composition rather than temperature is controlled.

b) direct temperature and pressure control if the scheme includes

-te/r, te/h or ts/h with the same comments for composition control.

c) indirect temperature and pressure control if the scheme includes;

- te/d , te/dv , $te/compressor\ recycle$ or ts/b

d) mass balance difficulties if the scheme includes both

a/r and lb/h (ie. no exit stream on level control) when there is a liquid distillate only.

a/r and lb/h and dv isn't on pressure control when there is a two phase distillate ($dv =$ a vapour distillate product).

lb/h and dv is not on pressure control when there is a vapour distillate only.

4.1 Two Composition Control

If two composition control is specified the program requests the following input data;

- 1) Whether the distillate is the smaller of the two product flows
- 2) Bottoms composition of the lower boiling point component
- 3) Tops composition of the lower boiling point component
- 4) Feed composition of the lower boiling point component
- 5) The number of theoretical trays
- 6) The reflux ratio

and uses it to calculate relative gains at the (1,1) position of a 2 x 2 relative gain array featuring top and bottom composition and two manipulated variables. The calculation is only valid for binary separations at this point. The program finds the possible pressure control alternatives from the input data and presents the most likely of the 8 possible two composition control schemes with comments to the user, along with the pressure control possibilities (The program only forms loops using the rules for pressure control in the Formloops section of the program in the two composition case).

The rules for selecting two composition control alternatives are;

- 1) If the distillate is the smaller product flow and $Rsv < 5$ then the SV scheme is a possibility

- 2) If the distillate is the smaller product flow and $R_{sv/b} < 5$ then the SV/B scheme is an alternative
- 3) If the distillate is the smaller product flow, $R_{sv/b} > 5$ and $R_{dv/b} > 0.9$ then the DV/B scheme is an alternative
- 4) If the distillate is the smaller product flow and $R_{sv/b} > 5$ and $R_{dv} > 0.9$ then the DV scheme is an alternative
- 5) If the distillate is the smaller product flow and $R_{sv/b} > 5$ and $R_{sd} > 0.9$ then the RSD scheme is a possibility
- 6) If the bottoms is the smaller product flow and $R_{sv/b} > 5$ and $R_{sb} > 0.9$ then the SB scheme is a possibility
- 7) If the bottoms is the smaller product flow and $R_{sb/l} < 5$ then the RSB/L scheme is a possibility
- 8) If $R_{lv} < 5$ then the RLV scheme is a possibility.

The program also outputs the allowable pressure control alternatives when two composition control is required and recommends reflux temperature control if there is a two phase distillate.

Appendix AIV - Rule Base For The Heat Exchanger Expert System

This appendix describes, in detail, the rules in the Heat Exchanger Expert System beginning with the "Input" part of the program.

1.0 Input

In the first instance, for a heat exchanger without a phase change, the questions are started by the sub goal call "Input". There is an initial group of questions to establish;

- i) Whether the controlled stream is heated or cooled
- ii) Whether the heating/cooling stream is a process or utility stream
- iii) What the characteristics of the disturbances likely to hit the exchanger are. A menu of choices is displayed and the user selects an option. The possibilities are "slow/small", "fast/small", "slow/large" and "fast/large". The speed refers to the rate of change of the upset variable's value and the size to the magnitude of the change.

The next series of questions follows the same format as a similar set in the distillation program and all require yes (y), no (n), why (?) answers. As the database builds up, queries are only asked if the correct conditions exist there.

- i) Whether the major disturbance is in a flow (asked if the disturbances are fast and of any magnitude) then the program requires the user to identify the disturbed flow from a menu (either the controlled stream, the heating/cooling stream or the cooling water stream).
- ii) Whether the utility stream is cooling water (asked only if the cooling stream is identified as a utility stream).
- iii) Whether there is a bypass around the exchanger on the controlled stream.

AIV-2

- iv) Whether there is a bypass around the exchanger on the other stream (only asked if there is no bypass on the controlled stream. As there is only one control objective assumed there should only ever be one bypass around the exchanger)
- v) Whether the controlled stream can be throttled (only if a bypass isn't available)
- vi) Whether the heating/cooling stream which is not cooling water can be throttled (only if there is no bypass or the temperature controlled stream cannot be throttled)
- vii) Whether the cooling water temperature will remain below 50 degrees C if it is throttled for control (Only if there is no bypass and the controlled stream cannot be throttled and the utility stream is cooling water)
- viii) Whether there is sufficient driving force for a single bypass valve. This is required to prevent valve saturation (only if a bypass is available)
- ix) Whether the operating temperature is $> 260^{\circ}\text{C}$ or the valve pressure drop is high (only if a bypass is available)

If the heat exchanger has a completely condensed stream the questions are started by the subgoal call "sh_input". There is only one initial question;

- i) Establish the disturbance characteristics (same format as the question for heat exchangers without a phase change)

The following set of questions are accessed using the same code as the other yes/no/why questions;

- i) If the expected disturbances are slow establish whether the exchanger is designed to run with a condensate level
- ii) If the expected disturbances are slow and large then establish whether the exchanger will be mounted vertically
- iii) If the disturbances are fast establish whether there is a bypass on the controlled stream
- iv) If the disturbances are fast and there is no bypass available establish whether the disturbances are in the steam header pressure
- v) If the disturbances are fast and there is no bypass available then establish whether there are significant disturbances in controlled stream temperature and flow

vi) Establish whether the condensing pressure is less than the condensate removal pressure.

vii) If the condensing pressure isn't less than the condensate removal header pressure then establish whether the exchanger is oversized for some operating condition.

The questions relating to bypass design (numbers viii and ix in the heat exchanger without phase change section) are also asked for this class of heat exchanger.

2.0 Control Methods

There are rules collected together to suggest control methods for the two groups of heat exchanger.

2.1 Heat Exchangers Without Phase Change

The rules in this class are as shown;

- i) If the expected disturbances are slow and the target stream can be throttled then control temperature by throttling the temperature controlled stream.
- ii) If the disturbances are slow and the target stream cannot be throttled and cooling is by cooling water that can be throttled and still remain less than 50°C then control temperature by throttling cooling water flow.
- iii) If the disturbances are slow and the target stream cannot be throttled and the other stream exchanging heat is not cooling water and this stream can be throttled then control temperature by throttling this stream.
- iv) If the target stream has a bypass then control temperature by bypassing some of this stream.
- v) If there is a bypass on the other stream (cooling water) then bypass the cooling water for control.
- vi) If there is a bypass on the other stream, which isn't a cooling water stream, bypass this stream for control.

vii) If the disturbances are fast and expected in the controlled stream flow and the controlled stream can be throttled then control temperature by throttling the controlled stream improving response with a secondary cascade loop.

viii) If the disturbances are fast and expected in the cooling water flow and the target stream cannot be throttled but the cooling water stream can then control temperature by throttling the cooling water stream improving response with a secondary cascade loop.

ix) If the disturbances are fast and expected in the heating/cooling stream (not cooling water) and the target stream cannot be throttled but the heating/cooling stream can then control temperature by throttling the heating/cooling stream improving response with a secondary cascade loop.

x) If the disturbances are fast and the controlled stream can be throttled then use a feed-forward system manipulating the controlled stream flow.

xi) If the disturbances are fast and the cooling water stream can be throttled then use a feedforward control system manipulating the cooling water flowrate.

xii) If the disturbances are fast and the other stream (not cooling water) can be throttled then use a feedforward system manipulating this flowrate.

xiii) If there is no bypass on the exchanger and no control method has been suggested for it try again adding a bypass to the exchanger as this adds an extra possibility for control.

2.2 Complete Phase Change Heat Exchangers

The rules for this class of exchangers are;

i) If the disturbances are slow and small and the exchanger is designed to run with a condensate level and the condensing pressure is not less than the condensate removal pressure then control temperature by adjusting condensate level in the exchanger.

ii) If the disturbances are slow and small and the exchanger cannot be controlled using the methods from (i) then control temperature by throttling steam flow to the exchanger.

iii) If the disturbances are fast and occur in the steam header pressure and there are not significant disturbances in controlled stream temperature/flow then control temperature by throttling steam to the exchanger improving response with a secondary cascade loop.

- iv) If the disturbances are fast and there is a bypass available then control temperature using a bypass valve held on control by adjusting steam valve position.
- v) If the disturbances are fast and significant upsets are expected in controlled stream inlet flow and temperature then control temperature using a feed forward system adjusting the steam valve.
- vi) If the disturbances are slow and large and the exchanger is designed to run with a condensate level and is mounted vertically and the condensing pressure is not less than the condensate removal header pressure then control temperature by varying condensate level.
- vii) If the disturbances are slow and large and the exchanger cannot be controlled using the method in (vi) then control temperature by throttling the steam valve.

There are a series of rules for establishing condensate removal methods;

- i) If the disturbances are small and the condensing pressure is not less than the condensate removal pressure and the exchanger is not oversized for some operating conditions then use a normal steam trap.
- ii) If the condensing pressure is less than the condensate removal pressure then use a pumping trap.
- iii) If disturbances are large and the exchanger is not oversized for some operating condition and the condensing pressure is not less than condensate removal pressure then use a drainer trap or a level controlled vessel.
- iv) If the exchanger is oversized for some operating condition and the condensing pressure is not less than the condensate removal pressure then use a drainer trap or a level controlled vessel.

2.3 Bypass Design Rules

These rules apply to both types of heat exchanger if there is a bypass around the exchanger;

- i) If a bypass is available and there is sufficient pressure drop then use a single two-way valve in the bypass.

AIV-6

ii) If a bypass is available and there is not sufficient pressure drop and the heat exchanger does not work at temperatures in excess of 260°C or with a high pressure drop across the valves then use a three way bypass valve.

iii) If a bypass is available and there isn't sufficient pressure drop and the exchanger works at high temperature (ie. > 260°C) or with a large pressure drop across the valves then use two two-way valves.

Appendix AV - Rule Base For The Reactor Expert System

1.0 Knowledge in module REACT1

The program is written in several sections the first is, as usual, the input section. The programming style closely follows that used in previous programs.

1.1 Input

There are a series of questions requiring either the selection of one of a number of options or a single word reply that is added to the data base (the same style as the previous programs for distillation and heat exchange);

i) Establish if the reactor is either a CSTR or belongs to the class of tubular/fixed bed reactors.

the next questions require that the reactor is a CSTR

i) Establish if the reaction is exothermic or endothermic or exothermic with the required feed preheat exceeding the heat release by the reaction.

ii) If the reactor is exothermic establish whether the heat evolution/volume is small, medium or large.

iii) If the reaction is exothermic and doesn't have a large heat evolution/volume establish whether the CSTR has a jacket, a coil or no attached heat exchange as a means of cooling.

iv) If the heat evolution/volume is large then the CSTR must have no attached heat exchange.

v) If the cooling is by jacket establish whether the cooling fluid is cooling water, heat transfer fluid, steam generation or cooling water with steam for initiating reaction.

The next series of questions all require a yes / no / ? answer;

- i) If the reaction is exothermic and the cooling fluid in the jacket is not boiling water then establish whether variations in coolant supply temperature are expected.
- ii) If the heat evolution/vol is medium for an exothermic reaction establish whether significant lags are introduced by temperature measurement, heat removal or reaction mass.
- iii) If the reaction is unstable and cooling of an exothermic reaction is by a jacket and the cooling fluid is cooling water or heat transfer fluid then establish if initial heating is required for reaction ignition.
- iv) If the CSTR has no attached cooling method establish if a catalyst feed is available to the reactor.
- v) If the CSTR has no attached cooling method establish if the reaction material is non-corrosive and neither a slurry nor a polymer
- vi) If the reaction is unstable and the reactor is cooled by a jacket using heat transfer fluid and does not require initial heating then establish if the fastest possible response is needed to temperature fluctuations.
- vii) If the heat evolution/vol is large establish whether the reactor has a condensing/refluxing system on reaction vapour.
- viii) If the reactor has a condensing/refluxing system on reaction vapour then establish if there is a continuous vapour product from the reactor.
- ix) If there is a continuous vapour product from the reactor then establish if there is an inerts feed to the partial condenser.

1.2 Rules for Reactor Stability

There are a set of rules for assessing the stability of the reactor which is important in the selection of temperature control methods;

- i) If the reaction heat evolution rate/vol is medium and there are no significant lags then the reactor is stable.
- ii) If the reaction heat evolution rate/vol is small then the reactor is stable.
- iii) If the reaction heat evolution rate/vol is medium and there are significant lags then the reactor is unstable.

Ultimately the program should use a numerical test for stability but these simple rules are sufficient at this stage in the program's development.

1.3 Temperature Control Method Selection Rules

There are twenty rules for selecting temperature control methods for exothermic reactions in a CSTR. The inference engine checks the conditions in all rules in the same way as the heat exchanger and distillation programs;

- i) If the reactor is cooled using a coil and the reaction is stable and there are no variations in coolant temperature expected then control temperature by throttling the coolant flow through the coil.
- ii) If the reactor is cooled using a coil and the reaction is stable and variations in coolant temperature are expected then control temperature by throttling the coolant flow through the coil (using cascade onto a secondary coolant temperature control loop).
- iii) If the reactor is cooled using a jacket and the reaction is stable and heat transfer fluid or cooling water is used and no variations in coolant temperature are expected then control temperature by throttling flow to the jacket.
- iv) As for rule (iii) except variations in coolant temperature are expected then control temperature by throttling the flow to the jacket and cascade the primary loop onto a secondary coolant temperature control loop.
- v) If the reactor is exothermic and stable then control temperature by varying feed temperature
- vi) If the reactor is cooled using a cooling water jacket with additional steam for start-up (reaction ignition) and the reaction is stable then control temperature by throttling cooling water flow and use steam in the jacket for initial heating.
- vii) If the reactor has no attached heat exchange and a catalyst feed is available then control temperature by varying catalyst feed rate.
- viii) If the reactor is cooled using a jacket and the reaction is unstable (ie. medium heat release with time lags) and cooling water is used in the jacket and the reaction doesn't require initial heating then control temperature using a pumparound through the jacket (make-up with cold coolant under temperature control and relieve hot coolant under pressure control).

ix) If the reactor is cooled using a jacket and the reaction is unstable and cooling water or heat transfer fluid is used in the jacket and initial heating is required then control temperature using a pumparound as before with a steam heated exchanger in the circuit for initial heat-up.

x) If there is no attached heat exchange and the reaction material is non-corrosive and neither a slurry nor a polymer and variations in coolant temperature are not expected then control temperature by circulating reaction mixture through an external heat exchanger.

xi) As for rule (x) but variations in coolant temperature are expected then use the cascade version of (x).

xii) If the CSTR has no attached cooling method and the heat evolution rate isn't large and the reaction mixture is corrosive etc. and there is no catalyst feed available then send a message to the user advising him to try a CSTR with a jacket instead (This rule traps an exceptional circumstance and warns the designer of it).

xiii) If the reaction is unstable and initial heating is not required and the reactor has a jacket and the reactor is cooled using heat transfer fluid and fast dynamics are required then control temperature using a bypass across an exchanger in a coolant recirculation loop.

xiv) As for (xiii) except fast dynamics aren't required then control temperature by throttling the cooling flow to an exchanger in the coolant circulation loop.

xv) If the reactor has a jacket and steam is generated in it for cooling and the reaction is unstable then control temperature by varying the steam generation pressure in the jacket.

xvi) If the reaction has a large heat release/vol and reaction vapour is condensed and recycled and there is no continuous vapour product or inerts feed to the condenser then control temperature by varying condenser cooling rate.

xvii) If the reaction has a large heat evolution rate/vol a continuous vapour product, condensed and recycled reaction vapour and an inerts feed to the condenser then control temperature using a split-range controller on vapour product release and inerts feed.

xviii) If the reactor is cooled by a coil and is unstable then try a reactor with a jacket cooling method instead (this rule traps an exceptional circumstance for the designer).

xix) if the feed preheat exceeds the heat released by reaction then control temperature using a split range controller on heat input and cooling rate.

There is a single rule for temperature control of endothermic reactors;

xx) If the reaction is endothermic then place the heat source on flow control and monitor the reaction temperature.

2.0 Knowledge in the Module CSTR

This module of the program establishes inventory and condition/composition control systems for Continuous Stirred Tank Reactors. The first part of the program is the input section.

2.1 Input

There are a number of initial questions requiring either selection of one of a number of options or a single word answer;

- i) Establish whether the reactor is a single pass system or part of a recycle loop.
- ii) If the reactor is single pass then establish whether the feed is a single liquid feed, two liquid feeds or a gas feed and a liquid feed.
- iii) If the reactor is a recycle type then establish if the feed is two pure component liquids or two pure liquids moderated by a recycled solvent.
- iv) If the reactor is single pass with two liquid feeds establish if one or both feeds are available for control.
- v) If the reactor is fed with a gas and a liquid then establish whether the product is gaseous or liquid.

The next series of questions require a yes / no / ? answer in the same style as the other programs;

- i) If the reaction is single pass establish whether conversion is a control objective
- ii) If the reactor is single pass and conversion is not a control objective establish whether exit composition is a control objective.
- iii) If the reactor is single pass establish whether reactant feed composition varies.

- iv) If the reactor is single pass with a single liquid feed and composition is a control objective and there are variations in feed composition expected then establish whether the feedrate is available for control.
- v) If the reactor is single pass and composition is a control objective and variations are expected in feed composition then establish whether catalyst feed is available as a manipulated variable.
- vi) If the reactor is a recycle type with two pure component feeds establish whether one of the reactants is in excess.
- vii) If the reactor is a recycle type with two pure component feeds and an excess of one of the reactants establish whether storage is available for the excess reactant for recycle.
- viii) Same conditions as (vii) then establish whether the loop separation stage returns uncontaminated excess reactant.
- ix) If the reactor is a recycle type and the reaction is moderated by a solvent then establish whether storage is available for the recycled solvent.
- x) Same conditions as (ix) establish whether some reactant is recycled with the solvent after separation from the products.

2.2 Control Method Selection Rules

There are a series of rules scanned by the program for selecting the CSTR inventory control ;

- i) If the reactor is single pass and does not have mixed phase feeds then the product should be on level control.
- ii) If the reactor is single pass with mixed phase feeds and the product is gaseous then the gaseous feed can be added on flow control, the liquid feed on level control and the gaseous product removed on pressure control. There should be a manual blowdown to remove inerts from the liquid.
- iii) If the reactor is single pass with mixed phase feeds and a liquid product then the liquid is added on flow control, the gas feed on pressure control and the liquid product removed on level control. There should be a manual purge to remove inerts from the gas space.

iv) If the reactor is a recycle type with two pure component feeds and an excess of one component and storage is available for the recycled reactant and the recycle separation stage returns pure reactant then feed the reactants on flow control. Make up the consumed excess reactant on level control to the recycle storage. The product should be on level control.

v) If the reactor is a recycle type and the reaction is moderated by a solvent and solvent storage is available then feed the reactants on flow control with ratio control between them. Recycle the solvent from storage on flow control. The product should be on level control.

There are a further set of rules for determining composition and condition control of a CSTR;

i) If the reactor is single pass and there is one feed to the reactor and the control objective is conversion then place the reactor feed on flow control.

ii) If the same conditions as in rule (i) apply except that the control objective is now composition and no variations in feed composition are expected then place the reactor feed on flow control.

iii) If the reactor is single pass and there is one feed to the reactor and the control objective is composition and feed composition varies and feed rate is available as a manipulation then vary feed rate to control residence time.

iv) If the reactor is single pass and there are two liquid feeds and both feeds are available for control and the objective is conversion then place the two feeds on flow control with ratio control between them.

v) If the reactor is single pass and there are two liquid feeds and both feeds are available for control and the objective is composition control and there are no expected variations in feed composition then place the two feeds on flow control with ratio control between them.

vi) If the reactor is single pass and there are two liquid feeds and the objective is composition control and there are variations in feed composition then place the two feeds on flow control with ratio control between them and trim the ratio using an exit composition controller

vii) If the reactor is single pass and there are two liquid feeds and only one is available for control and the objective is conversion or composition then control composition by ratioing the controllable feed to the other one and change the setpoint on a level

controller on the product to control spacetime. Override this signal with a low signal selector and a remote controller if the level exceeds a limit value.

viii) If the reactor is single pass and there is a gas and liquid feed and conversion is the objective then the proposed inventory control scheme also maintains conversion.

ix) If the reactor is single pass and there is a gas and liquid feed and composition is the objective and there are no expected disturbances in feed composition then the proposed inventory control scheme also maintains conversion.

x) If the reactor is single pass and the objective is composition control and variations are expected in feed composition and the reaction is stable then control composition by varying reaction temperature.

xi) If the reactor is single pass and the objective is composition control and variations are expected in feed composition and catalyst feed is not being used to control temperature then control composition by varying catalyst feedrate.

xii) If the reactor is a recycle reactor with the reactants on flow control, the consumed excess reactant made up on level control to the recycle storage and the product on level control then this will maintain conversion/composition.

xiii) If the reactor is a recycle type with no recycle of reactant in the solvent then the inventory control scheme will maintain conversion/composition.

xiv) If the reactor is a recycle type with the reactants fed on flow control with ratio control between them, solvent recycled from storage on flow control, the product on level control and reactant is recycled with the solvent then trim the ratio controller on the two feeds using feedback from a composition controller.

3.0 Knowledge in The Module TUBFBR

The final part of the reactor expert system handles control of tubular and fixed bed reactors. Once again there is the standard Input section.

3.1 Input

There are five initial questions requiring selection of one of a number of options;

i) Establish whether the reaction is endothermic or exothermic

- ii) Establish whether the heat evolution/volume is small, medium or large
- iii) Establish whether the reactor is single pass or part of a recycle loop.
- iv) Establish whether there are one or two feeds to the reactor.
- v) Establish if the reactor is a jacketed pipe, a heat exchanger type reactor, a multi fixed bed reactor or a fired heater.
- vi) If the reactor is exothermic and the reactor is a jacketed pipe or a heat exchanger type reactor (with or without catalyst in the tubes) establish whether the cooling fluid is a coolant or boiling water.

The next set of questions require yes / no / ? answers;

- i) If the heat evolution/volume is medium establish whether significant lags are introduced by temperature measurement, heat removal or reaction mass.
- ii) If the reaction is exothermic and the reactor is a multibed fixed bed reactor then establish whether the temperature of the feed before preheating is \ll reaction temperature.
- iii) If the reaction is exothermic and is cooled either by liquid coolant or steam raised in the jacket then establish whether catalyst feed is available as a manipulated variable.
- iv) If the catalyst feed is available as a manipulated variable then establish whether the feed flowrate or composition is variable.
- v) If the reactor is part of a recycle loop establish whether a purge bleed is available as a manipulated variable.
- vi) If a purge bleed is available establish whether inerts enter the recycle loop with the feed.
- vii) If inerts enter with the feed establish whether the inerts concentration is variable.
- viii) If the inerts concentration is variable establish whether the purge flowrate is measured.
- ix) If the reactor is a tubular or heat exchanger type with a coolant in the jacket or shell then establish whether variations are expected in the flow or temperature of the coolant.

3.2 Stability Rules

- i) If a reaction has a medium heat release and there are no significant lags then it is stable
- ii) If a reaction has a small heat release then it is stable.

3.3 Temperature Control Rules for Tubular and Heat Exchanger Reactors

There are a series of rules for establishing temperature control of the reactor checked in the same way as in the previous modules;

- i) If the reaction is exothermic and the reactor is a jacketed pipe or heat exchanger with coolant flow in the jacket and the reactor is stable then control temperature by throttling coolant flow to the jacket.
- ii) If the reactor is controllable by throttling coolant flow and variations are expected in coolant flow and temperature then cascade the primary loop onto a secondary coolant temperature control loop.
- iii) If the reaction is exothermic and the reactor is a jacketed pipe cooled by steam raised in the jacket and the reaction heat evolution rate is not large then control temperature by adjusting the steam pressure in the jacket.
- iv) If the reaction is exothermic and the reactor is a heat exchanger type cooled by steam raised in the shell then control temperature by varying steam pressure in the shell.
- v) If the reaction is endothermic and the reactor is a fired heater then control outlet temperature by varying furnace firing rate.

3.4 Temperature Control Rules for Fixed Bed Reactors

- i) If the reaction is exothermic and the reactor is a multi bed fixed bed reactor with a low temperature feed before preheating then control bed inlet temperatures by varying quench flowrate to the beds.
- ii) If the reaction is exothermic and the reactor is a multi bed fixed bed reactor and the feed is not low temperature then control bed inlet temperature by varying inter-bed heat exchanger cooling rate.

iii) If the reaction is endothermic and the reactor is a multi-bed fixed bed reactor then control temperature by varying inter-bed heat exchanger heating rate.

3.5 Rules for Exceptional Circumstances

i) If the reaction is exothermic and the heat evolution is large and the reactor is a tubular or heat exchanger type cooled using coolant flow in the jacket or shell then try again with a heat exchanger reactor cooled by steam raised in the jacket/shell.

ii) If the reaction is exothermic with a large heat evolution and the reactor is a jacketed pipe with steam raised in the jacket then try again with a heat exchanger reactor cooled by steam raised in the jacket.

iii) If the reaction is exothermic with a medium heat release and significant lags are present and the reactor is a jacketed pipe or heat exchanger type cooled by coolant then try again with a heat exchanger type reactor cooled by steam raised in the jacket.

iv) If the reaction is endothermic and the reactor is a tubular or heat exchanger type then there are no rules to support this choice. Try a fixed bed alternative.

There is also a rule for recommending reaction rate control;

i) If a recommended control method is to control temperature by varying jacket steam pressure and a catalyst feed is available as a manipulated variable and the feed composition or flowrate is variable then control reaction rate by varying catalyst feed to the reactor.

3.6 Feed and Pressure Control Rules in TUBFBR

There are also a number of rules to suggest inventory control techniques for this class of reactor;

i) If there is a single feed to the reactor place the feed on flow control

ii) If there are two feeds to the reactor then place both feeds on flow control with ratio control between the two.

iii) If the reactor is part of a recycle loop and a purge bleed is available then control pressure in the recycle loop by varying bleed rate.

AV-12

iv) If the reactor is part of a recycle loop and a purge bleed is available and there are inerts with varying concentration in the feed to the loop and the purge bleed flow is measured then control loop pressure by varying bleed rate and alter the flowrate of feed to the loop in response to changes in purge flowrate.

Appendix AVI - TurboProlog Listing For The Distillation Expert System

```
code = 2500
domains
  var = symbol
  type= symbol
  state= string
  varlist=var*
  num=real
database
  pair(var, var)
  con(varlist)
  manip(var)
  cond(type)
  small(var)
  col(var)
  flow(var)
  comp(var)
  quality(var)
  pvapour(var)
  top(var)
  sidedraw(var)
  tower(var)
  destn(var)
  drum(var)
  inert(var)
  rdv(num,varlist)
  rlv(num,varlist)
  rsv(num,varlist)
  rdvb(num,varlist)
  rsb(num,varlist)
  rsbl(num,varlist)
  rsvb(num,varlist)
  rsd(num,varlist)
  count(integer)
  small_d
  small_b
predicates
  run
  prodsiz
  man(var)
  cleanup
  input
  fcond
  inform
  convar
  convar1
  form_combinations
  controllable
  createloops
  testloops(varlist,varlist,varlist)
```

AVI-2

```
formloops
pressure
acclevel(varlist)
botlevel(varlist)
reflux(varlist)
sdraw(varlist)
distillate(varlist)
bottoms(varlist)
heat(varlist)
toptemp(varlist)
bottemp(varlist)
top_quality(varlist)
bot_quality(varlist)
member(var,varlist)
append(varlist,varlist,varlist)
reverse(varlist,varlist)
reverse1(varlist,varlist,varlist)
del(var,varlist,varlist)
check_head(varlist,varlist)
check_head1(varlist,varlist)
check_head2(varlist,varlist)
check_head3(varlist,varlist)
check_head4(varlist,varlist)
check_head5(varlist,varlist)
twocomp(varlist)
massbalance(varlist,varlist,integer)
vfcontrol(varlist,varlist,integer)
directTP(varlist,varlist,integer)
indirectTP(varlist,varlist,integer)
dtpcomment(varlist,integer)
indtpcomment(varlist,integer)
indtpcomment1(varlist,integer)
comment(varlist)
relgain
increment
config
repeat
askable(var, state)
ask_user(var, state)
check_condition(var)
check_reply(var,var)
explanation(var)
action(var)
vap_dist
two_phase
n_t_c
d_s_ph
r_t (varlist)
set_up_window
goal
run.

clauses

man(h).
man(r).
man(b).

/* This program section controls program operation */

run:- set_up_window,repeat,form_combinations,
```

```

        write("Would you like another consultation [y/n] ?"),
        readchar(Reply),write(Reply),nl,Reply= 'n'.

set_up_window:- makewindow(1,71,7,"Distillation Control Systems",0,0,25,80),
                clearwindow.

form_combinations:-
asserta(con([a,lb,p])),input,relgain,formloops,controllable,createloops,
                cleanup,!.
form_combinations.

/* This section of the program handles input data for */
/* a particular column case.*/

input:- write(" Enter the name of this case "),readln(Var),nl,
        writedevic(printer),write(" DISTILLATION CONTROL ADVISORY
        SYSTEM. CASE ",Var),nl,
        writedevic(screen),fcond,inform,convar,con(A),not(twocomp(A)),convar1,!.
input.
fcond:-write(" Is the feed under accurate flow control [y/n] ? "),
        readln(Ch),nl,Ch= y,asserta(flow(stable)),writedevic(printer),
        write(" Feed flow is stable "),nl,writedevic(screen),fail.
fcond:-write(" Is the feed flowing from a well mixed storage tank or does it have only
        a slowly varying composition [y/n] ? "),
        readln(Ch),nl,Ch= y,asserta(comp(stable)),writedevic(printer),
        write(" The feed composition is stable "),nl,writedevic(screen),fail.
fcond:-write(" Is the feed quality (percent vapourized, subcooling) accurately
        controlled [y/n] ? "),readln(Ch),nl,Ch= y,
        asserta(quality(stable)), writedevic(printer),
        write(" The feed quality is constant "),nl,writedevic(screen),fail.
fcond.

inform:- write(" Does the column have a [steady/zero/intermittent] vapour product ? "),
        readln(Var),nl,asserta(pvapour(Var)),writedevic(printer),write(" The column
        has a ",Var," vapour product "),nl,writedevic(screen),fail.
inform:- pvapour(steady),write(" Is the vapour product sent to a lower or compressed
        to a higher pressure [low/high] ? "),readln(Var), nl,asserta(destn(Var)),
        writedevic(printer),write(" The vapour product is sent to a ",Var,"
        pressure"),nl,writedevic(screen),fail.
inform:- write(" Is the column condenser air,cooling water or process cooled
        [air/water/process] ? "),readln(Var),nl,asserta(cond(Var)),writedevic(printer),
        write(" The column condenser is ",Var," cooled "),nl,writedevic(screen),nl,
        fail.
inform:- askable(Id,Question),ask_user(Id,Question),fail.
inform:- tower(draw),write(" Is the sidedraw a liquid or vapour [liquid/vapour] ? "),
        readln(Var),nl,asserta(sidedraw(Var)),writedevic(printer),
        write(" The sidedraw is a ",Var),nl,writedevic(screen),fail.
inform.

ask_user(Id,Question):- check_condition(Id),
        repeat,
        write(Question),
        readln(Reply),
        check_reply(Id,Reply),!.

check_reply(Id,why):- explanation(Id),!,fail.
check_reply(Id,y):- writedevic(printer),action(Id),writedevic(screen).
check_reply(_,n).

```

```

explanation(pf):- write(" If the condenser runs with a liquid level then", " level and
therefore", "\n", " heat transfer area can be varied using ", "condensate flow.
Pressure control", "\n", " is then possible by condensate rate"),nl.
explanation(df):- write(" If the reflux drum is flooded then reflux ", "drum level isn't a
necessary", "\n", " control objective and pressure can be ", "controlled using
reflux rate", "\n", " (if the condenser runs partially flooded)"),nl.
explanation(ab):- write(" Two pressure control methods require that ", "this condition is
satisfied;", "\n", " 1) If condensate is controlling ", " pressure then static head
must provide", "\n", "sufficient driving force to cause it to flow ", " into the
reflux drum ", "\n", "2) If vapour flow to the condenser is controlling", "
pressure (only with a", "\n", "partially flooded condenser) liquid level", "
changes with valve position", "\n", "because condenser pressure also
changes.", " The condenser operates", "\n", " at a lower pressure than the reflux
drum and must", " be above it and condensate", "\n", " must flow to the drum
because of the static head", " driving force"),nl.
explanation(liq):- write("In two pressure control methods ", "\n", "1) Pressure
controlled by vapour to condenser ", "\n", "2) Pressure controlled by a hot gas
bypass ", "\n", "The condensate must enter below liquid level to allow the
necessary ", "\n", "manometer effect between drum and condenser to operate
causing liquid ", "level", "\n", "to change in response to varying pressure
difference between", " condenser and drum."),nl.
explanation(sc):- write("If pressure is controlled by a hot gas bypass", " there must be a
temperature", "\n", " difference between column and reflux drum", " because
changes in delta T cause", "\n", " changes in delta P and therefore ", " changes in
the condenser level", "\n", "(partially flooded condenser required)"),nl.
explanation(inert):- write("Pressure can be controlled by venting the", " vapour product
or admitting inert gas", "\n", " to the drum. Two valves on split", "\n", " range
control are used"),nl.
explanation(equal):- write("Two pressure control methods require that ", "\n", " the
column and reflux drum operate at the same pressure; ", "\n", "1) Manipulation
of vapour flow to the condenser", "\n", "2) Manipulation", "of condensate ex the
condenser"),nl.
explanation(over):- write(" If the vapour overheads line is < 150 mm ", " then it is
economic to manipulate", "\n", " total overhead vapour flow to ", " control
pressure"),nl.
explanation(cool):- write("If column top temperature exceeds 50 degrees C", " and
cooling water flow is", "\n", " throttled for control then cooling water ", "
temperature may also exceed", "\n", " 50 degrees C and HX fouling would
increase"),nl.

check_condition(atmos):- not(pvapour(steady)).
check_condition(pf):- not(col(atmos)),not(pvapour(steady)).
check_condition(df):- pvapour(zero), manip(e).
check_condition(ab):- not(pvapour(steady)),not(drum(flood)),manip(e).
check_condition(liq):- not(col(atmos)),not(pvapour(steady)),not(drum(flood)).
check_condition(sc):- pvapour(zero),manip(e),not(drum(flood)).
check_condition(inert):- pvapour(intermittent).
check_condition(equal):- not(col(atmos)),not(drum(flood)),not(pvapour(steady)).
check_condition(bypass):- not(col(atmos)),not(drum(flood)),not(pvapour(steady)).
check_condition(dp).
check_condition(dist):- manip(d), not(drum(flood)).
check_condition(bot).
check_condition(over):- not(col(atmos)),not(drum(flood)),not(pvapour(steady)).
check_condition(cool):- not(col(atmos)),cond(water).
check_condition(lou):- cond(air).
check_condition(apb):- cond(air).
check_condition(le).
check_condition(he).
check_condition(sd).

action(atmos):- asserta(col(atmos)),write(" This is an atmospheric column "),nl.

```



```

action(pf):- asserta(manip(e)),write("The condenser runs partially flooded "),nl.
action(df):- asserta(drum(flood)),write(" The reflux drum is flooded "),nl.
action(ab):- asserta(cond(above)),write(" The condenser is above the reflux
      drum "),nl.
action(liq):- asserta(drum(below)),write(" The condensate enters below the liquid
      level "),nl.
action(sc):- asserta(cond(cool)),write(" The condensate is subcooled "),nl.
action(inert):- asserta(inert(av)),write(" Inert gas is available for pressure control"),nl.
action(equal):- asserta(tower(equal)),write(" There is an equalising line between
      column and reflux drum"),nl.
action(bypass):- asserta(tower(bypass)),write(" There is a bypass between column and
      reflux drum"),nl.
action(dist):- asserta(small(d)),write(" The distillate is too small for level control "),nl.
action(dp):- asserta(manip(d)),write(" There is a liquid distillate product "),nl.
action(bot):- asserta(small(b)),write(" The bottoms is too small for level control "),nl.
action(over):- asserta(manip(v)),write(" Total overhead is available for pressure
      control "),nl.
action(cool):- asserta(top(cool)),write(" Column top temperature is < 50
      degrees C "),nl.
action(lou):- asserta(cond(louvers)),write(" louvers are available for pressure
      control "),nl.
action(apb):- asserta(cond(fan_pitch)),write(" Fan pitch is available for pressure
      control "),nl.
action(le):- asserta(tower(lend)),write(" The column removes light ends "),nl.
action(he):- asserta(tower(hend)),write(" The column removes heavy ends "),nl.
action(sd):- asserta(tower(draw)),write(" The column has a sidestream product "),nl.

askable(atmos," Does the column operate at atmospheric pressure [y/n] ").
askable(pf," Is the condenser partially flooded [why/y/n] ").
askable(df," Is the reflux drum flooded or excluded [why/y/n] ").
askable(ab," Is the condenser located above the the reflux drum and the condensate
      pipework designed for gravity flow [why/y/n] ").
askable(liq," Does the condensate enter the reflux drum below the liquid level
      [why/y/n] ").
askable(sc," Is the condensate subcooled [why/y/n] ").
askable(inert," Is there a source of inert gas at higher than column pressure
      [why/y/n] ").
askable(equal," Is there a pressure equalising line between column and reflux drum
      [why/y/n] ").
askable(bypass," Is there a bypass between column and reflux drum [y/n] ").
askable(dp," Is there a liquid distillate product [y/n] ").
askable(dist," Does the distillate have too small a flow for reflux drum level control
      [y/n] ").
askable(bot," Does the bottoms have too small a flow for column level control
      [y/n] ").
askable(over," Does the column have an overheads line < 150 mm diameter
      [why/y/n] ").
askable(cool," Is the column top temperature < 50 degrees C [why/y/n] ").
askable(lou," Do the air cooler fans have air flow control louvers [y/n] ").
askable(apb," Do the air cooler fans have adjustable pitch blades [y/n] ").
askable(le," Does the tower remove a small lightends stream from the feed [y/n] ").
askable(he," Does the tower remove a small heavy ends stream from the feed [y/n] ").
askable(sd," Does the tower have a sidestream product [y/n] ").

```

/* This program section enters the required control objectives for a case. The default objectives are accumulator level, bottoms level and pressure */

```

convar:- flow(stable),quality(stable),comp(stable),!.
convar:-write(" Feed conditions are unstable, composition must be controlled "),nl,

```

```

write(" Is it required (and possible) to control composition in the top of the
column [y/n] ? "),readln(Ch),nl,Ch= y,con(A), asserta(con([qe|A])),
retract(con(A)),fail.
convar:- not(n_t_c),write(" Is it required (and possible) to control composition in the
bottom of the column [y/n] ? "),readln(Ch),nl,Ch= y,con(A),
asserta(con([qslA])), retract(con(A)),fail.
convar:-con(A),not(member(qe,A)),not(member(qs,A)),write(" Is top temperature a
required control objective [y/n] ? "),readln(Ch),nl,
Ch= y,asserta(con([telA])),retract(con(A)),fail.
convar:-con(A),not(member(qs,A)),not(member(qe,A)),not(member(te,A)),write(" Is
bottom temperature a required control objective [y/n] ? "),readln(Ch),nl,
Ch= y,asserta(con([tslA])),retract(con(A)),fail.
convar.
convar1:- manip(d),write(" Should liquid distillate rate be on flow control [y/n] ? "),
readln(Ch),nl,Ch= y,con(A),asserta(con([dlA])),retract(con(A)),fail.
convar1:- pvapour(steady),write(" Should the vapour product be on flow control
[y/n] ? "),readln(Ch),nl,Ch=y,con(A),asserta(con([dv|A])),retract(con(A)),fail.
convar1:- write(" Should bottoms rate be on flow control [y/n] ? "),readln(Ch),nl,
Ch= y,con(A),asserta(con([blA])),retract(con(A)),fail.
convar1:- write(" Should heat input be on flow control [y/n] ? "),readln(Ch),nl,
Ch= y,con(A),asserta(con([hlA])),retract(con(A)),fail.
convar1:- write(" Should reflux be on flow control [y/n] ? "),readln(Ch),nl,
Ch= y,con(A),asserta(con([rlA])),retract(con(A)),fail.
convar1:- tower(draw),write(" Should sidedraw rate be on flow control ?"),
readln(Ch),nl,
Ch= y,con(A),asserta(con([slA])),retract(con(A)),fail.
convar1:- two_phase,con(A),member(te,A),retract(con(A)),asserta(con([r_tlA])),
write(" There are both vapour and liquid distillate products so reflux
temperature\n",
"must also be controlled "),nl,fail.
convar1:- two_phase,con(A),member(qe,A),retract(con(A)),asserta(con([r_tlA])),
write(" There are both vapour and liquid distillate products so reflux
temperature\n","must also be controlled "),nl,fail.
convar1.

twocomp(A):- member(qe,A),member(qs,A).
vap_dist:- pvapour(steady),not(manip(d)).
two_phase:- pvapour(steady),manip(d).
n_t_c:- con(A),member(qe,A),vap_dist.
d_s_ph:- manip(d),not(pvapour(steady)).

/* This section of the program forms a database of all the allowable pairs */
/* of controlled and manipulated variables and prints out a list of */
/* the allowed pairs. */

formloops:- writedevic(printer),con(A),pressure,not(twocomp(A)),acclevel(A),
botlevel(A),reflux(A),sdraw(A),distillate(A),bottoms(A),heat(A),toptemp(A),
bottemp(A),top_quality(A),bot_quality(A),r_t(A),nl,
write(" LIST OF POSSIBLE CONTROL PAIRS "),nl,pair(X,Y),
write(" possible pair ", X, " ", Y),nl,fail.
formloops.
pressure:- col(atmos),asserta(pair(p,atmospheric)),fail.
pressure:- pvapour(steady),destn(high),asserta(pair(p,compressor_recycle)),fail.
pressure:- pvapour(steady),con(A),not(member(dv,A)),destn(low),
asserta(pair(p,dv)),fail.
pressure:- not(col(atmos)),not(pvapour(steady)),con(A),not(twocomp(A)),man(h),
asserta(pair(p,h)),fail.
pressure:- not(drum(flood)),cond(water),top(cool),not(two_phase),
asserta(pair(p,cooling_water_flow)),fail.
pressure:- cond(process),not(drum(flood)),not(two_phase),

```

```

    asserta(pair(p,process_cooling_flow)),fail.
pressure:-not(pvapour(steady)),manip(v),
    asserta(pair(p,overhead_vapour_reflux_vent_open)),fail.
pressure:-pvapour(zero),manip(v),
    asserta(pair(p,over_head_vapour_reflux_vent_closed)),fail.
pressure:-not(pvapour(steady)),manip(e),tower(equal),cond(above),drum(below),
    asserta(pair(p,vapour_to_condenser)),fail.
pressure:-not(pvapour(steady)),tower(equal),manip(e),cond(above),
    asserta(pair(p,condensate_ex_condenser)),fail.
pressure:-con(A),not(twocomp(A)),pvapour(zero),drum(flood),asserta(pair(p,r)),fail.
pressure:-cond(fan_pitch),not(two_phase),asserta(pair(p,fan_pitch)),fail.
pressure:-cond(louvers),not(two_phase),asserta(pair(p,louvers)),fail.
pressure:-not(pvapour(steady)),manip(e),tower(bypass),
    asserta(pair(p,condensate_ex_condenser_reflux_drum_pressure_controlled_on
        _bypass)),fail.
pressure:-pvapour(zero),tower(bypass),cond(cool),manip(e),not(cond(above)),
    drum(below),asserta(pair(p,hot_gas_bypass)),fail.
pressure:-pvapour(intermittent),inert(av),
    asserta(pair(p,split_range_reflux_vent_and_inert_bleed)),fail.
pressure.
acclevel(A):-not(drum(flood)),not(member(r,A)),man(r),asserta(pair(a,r)),fail.
acclevel(A):-not(drum(flood)),not(member(d,A)),manip(d),not(small(d)),
    asserta(pair(a,d)),fail.
acclevel(A):-not(drum(flood)),not(member(s,A)),sidedraw(liquid),
    asserta(pair(a,s)),fail.
acclevel(A):-drum(flood),del(a,A,A1),retract(con(A)),asserta(con(A1)),fail.
acclevel(_):-not(drum(flood)),vap_dist,cond(water),top(cool),
    asserta(pair(a,cooling_water_flow)),fail.
acclevel(_):-not(drum(flood)),vap_dist,cond(process),
    asserta(pair(a,process_cooling_flow)),fail.
acclevel(_):-not(drum(flood)),cond(louvers),vap_dist,asserta(pair(a,louvers)),fail.
acclevel(_):-not(drum(flood)),cond(fan_pitch),vap_dist,asserta(pair(a,fan_pitch)),fail.
acclevel(_).
botlevel(A):-not(member(b,A)),man(b),not(small(b)),asserta(pair(lb,b)),fail.
botlevel(A):-not(member(h,A)),man(h),asserta(pair(lb,h)),fail.
botlevel(_).
sdraw(A):-member(s,A),asserta(pair(s,s)),fail.
sdraw(_).
reflux(A):-member(r,A),man(r),asserta(pair(r,r)),fail.
reflux(_).
distillate(A):-member(d,A),manip(d),asserta(pair(d,d)),fail.
distillate(A):-member(dv,A),asserta(pair(dv,dv)),fail.
distillate(_).
bottoms(A):-member(b,A),man(b),asserta(pair(b,b)),fail.
bottoms(_).
heat(A):-member(h,A),man(h),asserta(pair(h,h)),fail.
heat(_).
toptemp(A):-member(te,A),manip(d),asserta(pair(te,d)),fail.
toptemp(A):-member(te,A),man(r),asserta(pair(te,r)),fail.
toptemp(A):-member(te,A),man(h),asserta(pair(te,h)),fail.
toptemp(A):-member(te,A),not(member(dv,A)),destn(low),asserta(pair(te,dv)),fail.
toptemp(A):-member(te,A),destn(high),asserta(pair(te,compressor_recycle)),fail.
toptemp(_).
top_quality(A):-member(qe,A),manip(d),asserta(pair(qe,d)),fail.
top_quality(A):-member(qe,A),man(r),asserta(pair(qe,r)),fail.
top_quality(A):-member(qe,A),man(h),asserta(pair(qe,h)),fail.
top_quality(A):-member(qe,A),not(member(dv,A)),destn(low),
    asserta(pair(qe,dv)),fail.
top_quality(A):-member(qe,A),destn(high),asserta(pair(qe,compressor_recycle)),fail.
top_quality(_).
bottemp(A):-member(ts,A),man(b),asserta(pair(ts,b)),fail.

```

```

bottemp(A):- member(ts,A),man(h),asserta(pair(ts,h)),fail.
bottemp(_).
bot_quality(A):- member(qs,A),man(b),asserta(pair(qs,b)),fail.
bot_quality(A):- member(qs,A),man(h),asserta(pair(qs,h)),fail.
bot_quality(_).
r_t(A):- two_phase,member(r_t,A),cond(water),top(cool),
        asserta(pair(r_t,cooling_water_flow)),fail.
r_t(A):- two_phase,member(r_t,A),cond(process),
        asserta(pair(r_t,process_cooling_flow)),fail.
r_t(A):- two_phase,member(r_t,A),cond(louvers),asserta(pair(r_t,louvers)),fail.
r_t(A):- two_phase,member(r_t,A),cond(fan_pitch),asserta(pair(r_t,fan_pitch)),fail.
r_t(_).

/* This section of the program checks whether the control objectives are */
/* controllable with the available manipulations */

controllable:- con(A),nl,write(" POSSIBLE CONTROL SCHEMES "),nl,
               not(twocomp(A)),
               write(" The control objectives are ",A),nl,testloops(A,[],_),!.
controllable:- con(A),twocomp(A),not(drum(flood)),pair(p,_),A=[qs,qe,a,lb,p],
               W=[qs,qe,a,lb],write(" The control objectives are ",W),nl,!.
controllable:- con(A),twocomp(A),drum(flood),A=[qs,qe,a,lb,p],W=[qs,qe,p,lb],
               write(" The control objectives are ",W),nl,!.
controllable:- write(" The control objectives aren't controllable with the available
               manipulated   variables "),nl,cleanup,writedevic(screen),fail.

/* This section of the program forms all the allowable combinations of */
/* pairs which make up control schemes for the column. */

createloops:- con(A),not(twocomp(A)),comment(A),asserta(count(1)),
               testloops(A,[],B1),
               reverse(B1,R),count(N),increment,
               not(massbalance(A,R,N)),not(vfcontrol(A,R,N)),
               not(directTP(A,R,N)),not(indirectTP(A,R,N)),
               write(" Case ",N," manipulated variables are ",R),nl,nl,fail.
createloops:- con(A),twocomp(A),asserta(count(1)),config,fail.
createloops:- pvapour(intermittent),not(drum(flood)),
               write(" All methods for pressure control except pressure ", "\n",
               " controlled by overheads flow with reflux vent open or splitrage ", "\n",
               " between reflux vent and inerts bleed must have reflux drum pressure", "\n",
               " controlled by vent flow "),fail.
createloops:- writedevic(screen).

comment(A):- not(member(te,A)),not(member(ts,A)),not(member(qs,A)),
               not(member(qe,A)),write(" The following are massbalance control schemes
               suitable because feed flow, ", "\n", "composition and quality are
               constant "),nl,!.
comment(_):- !.

testloops([],B1,B1):- !.
testloops([N|A1],B,B1):- pair(N,R),not(member(R,B)),
               append([R],B,L),testloops(A1,L,B1).

/* These are prolog utility routines required by the program */

member(R,[R|_]).
member(R,[_|Tail]):- member(R,Tail).

append([],B,B).
append([X|L1],List2,[X|L3]):- append(L1,List2,L3).

```

```

reverse(L,R):- reverse1(L,[],R).
reverse1([],R,R).
reverse1([H|R],W,R1):- reverse1(R,[H|W],R1).

del(X,[X|T],T):-!.
del(X,[Y|T],[Y|T1]):- del(X,T,T1).

repeat.
repeat:- repeat.

/* This is the output comment section of the program */

vfcontrol(A,R,N):- d_s_ph,check_head(A,R),!,check_head2(A,R),
    write("Case ",N," manipulated variables are ",R," representing v/f
    control."),nl,write("Control won't be as good as other temperature or
    composition schemes "),nl,nl.
vfcontrol(A,R,N):- d_s_ph,check_head1(A,R),!,check_head3(A,R),
    write("Case ",N," manipulated variables are ",R," representing v/f control."),
    nl,write("Control won't be as good as other temperature or composition
    schemes "),nl,nl.

directTP(A,R,N):- check_head3(A,R),!,dtpcomment(R,N).
directTP(A,R,N):- check_head2(A,R),dtpcomment(R,N).

indirectTP(A,R,N):- check_head4(A,R),check_head(A,R),!,indtpcomment1(R,N).
indirectTP(A,R,N):- check_head4(A,R),indtpcomment(R,N).

massbalance(A,R,N):- d_s_ph,!,check_head(A,R),check_head1(A,R),
    write("Case ",N," manipulated variables are ",R," leading to "),nl,
    write("massbalance problems. This scheme shouldn't normally be used"),nl,nl.
massbalance(A,R,N):- vap_dist,!,check_head1(A,R),not(check_head5(A,R)),
    write("Case ",N," manipulated variables are ",R," leading to "),nl,
    write("massbalance problems. This scheme shouldn't normally be used"),nl,nl.
massbalance(A,R,N):- two_phase,!,check_head(A,R),check_head1(A,R),
    not(check_head5(A,R)),
    write("Case ",N," manipulated variables are ",R," leading to "),nl,
    write("massbalance problems. This scheme shouldn't normally be used"),nl,nl.

config:- small_d,rsv(Rsv,R),Rsv < 5,count(N),
    write("Case ",N," manipulated variables are ",R),writef(" interaction
    coefficient Rsv= %6.2f\n",Rsv),
    write(" simpler version of the sv/b scheme"),nl,nl,increment,fail.
config:- small_d,rsvb(Rsvb,R),Rsvb < 5,count(N),
    write("Case ",N," manipulated variables are ",R),writef(" interaction
    coefficient Rsv/b= %6.2\n",Rsvb),
    write(" If Rsv/b is in the range 1-5 and distillate is the smallest product
    flow", "\n",
    "this is the recommended scheme"),nl,nl,increment,fail.
config:- small_d,rsvb(Rsvb,_),Rsvb > 5,rdvb(Rdvb,R),Rdvb > 0.9,count(N),
    write("Case ",N," manipulated variables are ",R),writef(" interaction
    coefficient Rdv/b= %6.2\n",Rdvb),
    write("If Rsv/b is > 5 and distillate is the smallest product flow and Rdv/b is
    0.9-1.0", "\n",
    "this scheme is recommended"),nl,nl,increment,fail.
config:- small_d,rsvb(Rsvb,_),Rsvb > 5,rdv(Rdv,R),Rdv > 0.9,count(N),
    write("Case ",N," manipulated variables are ",R),writef(" interaction
    coefficient Rdv= %6.2\n",Rdv),
    write("this is a material balance scheme possible if Rdv is 0.9-
    1.0"),nl,nl,increment,fail.
config:- small_b,rsvb(Rsvb,_),Rsvb > 5,rsb(Rsb,R),Rsb > 0.9,count(N),

```

```

        write("Case ",N," manipulated variables are ",R),writef(" interaction
coefficient Rsb= %6.2\n",Rsb),
        write("If bottoms flow is the smallest product flow and Rsb is 0.9-1.0 ", "\n",
        "this scheme is favoured"),nl,nl,increment,fail.
config:- small_b,rsbl(Rsbl,R),Rsbl < 5,count(N),
        write("Case ",N," manipulated variables are ",R),writef(" interaction
coefficient Rsb/l= %6.2\n",Rsbl),
        write("If Rsb is unfavourable and bottoms flow is the smallest product
flow", "\n", "and Rsb/l is in the range 1-5 this scheme is
recommended"),nl,nl,increment,fail.
config:- rlv(Rlv,R),Rlv < 5,count(N),
        write("Case ",N," manipulated variables are ",R),writef(" interaction
coefficient Rlv = %6.2\n",Rlv),write(" This is the conventional scheme usually
with poor interaction (Rlv large)"),nl,nl,increment,fail.
config:- small_d,rsvb(Rsvb,_),Rsvb > 5,rsd(Rsb,R),Rsb > 0.9,count(N),
        write("Case ",N," manipulated variables are ",R),writef(" interaction
coefficient Rsd= %6.2\n",Rsb),write("If Rsd is 0.9-1.0 and distillate is the
smallest product flow and Rsv/b and Rdv/b are unfavourable this scheme is
favoured"),nl,nl,increment,fail.
config:- not(drum(flood)),write(" Pressure can be controlled by the following
methods "), nl,pair(p,P),write("Pressure controlled by ",P),nl,fail.
config:- two_phase,write(" There are vapour and liquid distillate products so reflux
temperature\n", "must be controlled using condenser cooling rate. "),nl,fail.
config.

increment:- retract(count(N)),N1=N+1,asserta(count(N1)),!.

check_head([_l_],[_r_]):- !.
check_head([_lA1],[_lR1]):- check_head(A1,R1).

check_head1([_l_],[_h_]):- !.
check_head1([_lA1],[_lR1]):- check_head1(A1,R1).

check_head2([_qsl_],[_hl_]):-!.
check_head2([_qel_],[_hl_]):-!.
check_head2([_tel_],[_hl_]):-!.
check_head2([_tsl_],[_hl_]):-!.
check_head2([_lA1],[_lR1]):- check_head2(A1,R1).

check_head3([_qel_],[_rl_]):-!.
check_head3([_tel_],[_rl_]):- !.
check_head3([_lA1],[_lR1]):- check_head3(A1,R1).

check_head4([_qel_],[compressor_recycle_]):-!.
check_head4([_tel_],[compressor_recycle_]):-!.
check_head4([_qel_],[dv_]):-!.
check_head4([_tel_],[dv_]):-!.
check_head4([_qsl_],[bl_]):-!.
check_head4([_qel_],[dl_]):-!.
check_head4([_tel_],[dl_]):- !.
check_head4([_tsl_],[bl_]):- !.
check_head4([_lA1],[_lR1]):- check_head4(A1,R1).

check_head5([_pl_],[compressor_recycle_]):-!.
check_head5([_pl_],[dv_]):-!.
check_head5([_lA1],[_lR1]):- check_head5(A1,R1).
dtpcomment(R,N):- write("Case ",N," manipulated variables are ",R),nl,
        write(" This is a direct temperature or composition control scheme. "),nl,
        write(" It offers fastest response to control action and is better than indirect
T/P"),nl,write(" schemes when product feeds another column. "),nl,nl.

```

```

indtpcomment(R,N):- write("Case ",N," manipulated variables are ",R),nl,
    write(" This is an indirect temperature or composition control scheme. "),nl,
    write(" It has a slower response than direct T/P schemes but is better
    when"),nl,write(" bottoms/distillate flow is too small for level control. "),
    nl,nl.
indtpcomment1(R,N):- write("Case ",N," manipulated variables are ",R),nl,
    write(" This is an indirect temperature/pressure control scheme. ","\n"," It has
    a slower response than direct schemes but is better when\n"," bottoms or
    distillate is too small for level control.\n",
    " Coupling reflux with reflux drum level gives best response when upsets
    affect"),nl,write(" overhead cooling (eg. fin/fan cooler). "),nl,nl.

/* This section of the program cleans out the database in preparation for */
/* a new case */

cleanup:- retract(_),fail.
cleanup.

/* This section of the program computes the required*/
/* relative gains for different control configurations */

relgain:- con(A),twocomp(A),prodsiz,
    write(" Bottom composition (LC) ="),
    readreal(X),nl,
    write("Top composition (LC) ="),
    readreal(Y),nl,
    write("Feed composition (LC) ="),
    readreal(Z),nl,
    write("Number of theoretical stages ="),
    readreal(Ne),nl,
    write("Reflux ratio ="),
    readreal(RR),nl,
    D1=(Z-Y)/(Z-X),
    Sg=(Y*(1-Y))/(X*(1-X)),
    Ep=(Ne*Y*(1-Y))/(2*(Z*RR+1)*(Y-X)),
    Lf=(Sg-D1*Ep)/(1-Ep),
    Vf=(Sg-D1*Ep*(1+1/RR))/(1-Ep*(1+1/RR)),
    Fb=(Y-X)/(Y-Z),
    Lb=(Sg-D1*Ep*Fb)/(1-Ep*Fb),
    Vb=(Sg-D1*Ep*(1+1/RR)*Fb)/(1-Ep*(1+1/RR)*Fb),
    Rdv=1/(1-D1/Vf),asserta(rdv(Rdv,[h,d,r,b])),
    Rdvb=1/(1-D1/Vb),asserta(rdvb(Rdvb,[h_ratioed_to_b,d,r,b])),
    Rlv=1/(1-Lf/Vf),asserta(rlv(Rlv,[h,r,d,b])),
    Rsb=1/(1-Sg/D1),asserta(rsb(Rsb,[b,r_ratioed_to_d,d,h])),
    Rsv=1/(1-Sg/Vf),asserta(rsv(Rsv,[h,d_ratioed_to_r,r,b])),
    Rsbl=1/(1-Sg/Lb),asserta(rsbl(Rsbl,[b_ratioed_to_r,r_ratioed_to_d,d,h])),
    Rsvb=1/(1-Sg/Vb),
    asserta(rsvb(Rsvb,[h_ratioedto_b,d_ratioed_to_r_plus_d,r_plus_d,b])),
    asserta(rsd(Rsb,[d,r_ratioed_to_d,h,b])),!.
relgain.

prodsiz:- write(" Is the distillate the smaller product flow [y/n] ? "),
    readln(Ch),nl,Ch=y,asserta(small_d),!.
prodsiz:- asserta(small_b).

```


Appendix AVII - TurboProlog Listing For The Heat Exchanger Expert System

```
code=2048
domains
var=symbol
state=string
database
dist(var,var)
flow(var)
ff(var)
duty(var)
htcool(var)
target(var)
bypass(var)
cool(var)
dp(var)
htdp(var)
by(var)
con(var,var)
hload(var)
shx(var)
hp(var)
cr(var)
trap(var)
include "menu.pro"
predicates
input
shx_input
run
task
type
disturb
inform
inform1
repeat
cleanup
contype
sh_contype
bypas
byp_type
cr_type
cond_removal
con_meth
set_up_window
output_type(var,var)
output_type1(var)
output_type2(var)
askable(var,state)
askable1(var,state)
ask_user(var,state)
check_condition(var)
```

AVII-2

```

check_reply(var,char)
action(var)
explanation(var)
insert(integer)
insert1(integer)
hx_no_phase_change
steam_hx
proces(integer)
write_flow(integer)
goal
run.
clauses

```

```

run:- set_up_window,repeat,write("This expert system recommends control systems
    for","\n",
    "1) Heat exchangers with no phase change","\n",
    "2) Steam heated exchangers"),nl,
    write("Enter the heat exchanger type [1/2] "),readint(X),
    proces(X),write("Would you like another consultation [y/n/?] "),
    readchar(Reply),write(Reply),nl,Reply='n'.

```

```

proces(1):- hx_no_phase_change.
proces(2):- steam_hx.

```

```

hx_no_phase_change:- input,contype,bypas,con_meth,byp_type,cleanup.

```

```

steam_hx:- shx_input,sh_contype,bypas,con_meth,byp_type,cr_type,cond_removal,
    cleanup.

```

```

set_up_window:- makewindow(1,71,7,"Heat Exchanger Control Systems",
    0,0,25,80), clearwindow.

```

```

input:- write(" Enter the name of this case "),readln(Var),
    write(" HEAT EXCHANGER ADVISORY SYSTEM. CASE ",Var),nl,task,
    type,disturb,inform.

```

```

shx_input:- write(" Enter the name of this case "),readln(Var),
    write(" HEAT EXCHANGER ADVISORY SYSTEM. CASE ",Var),nl,
    disturb,inform1.

```

```

task:- write(" Is the controlled stream heated or cooled [heat/cool] "),
    readln(Var),asserta(duty(Var)).

```

```

type:- write(" Is the heating/cooling stream a process or utility stream [pro/ut] "),
    readln(Var),asserta(htcool(Var)).

```

```

disturb:- write(" Select expected disturbance characteristics with arrow key"),
    nl,menu(1,65,
    "slow/small",
    "fast/small",
    "slow/large",
    "fast/large"],
    Choice),insert(Choice),dist(A,B),write("The disturbance characteristics are
    ",A," and ",B),nl.

```

```

inform:- askable(Id,Question),ask_user(Id,Question),fail.
inform.

```

```

inform1:- askable1(Id,Question),ask_user(Id,Question),fail.
inform1.

```

```

ask_user(Id,Question):- check_condition(Id),
    repeat,write(Question),
    readchar(Reply),

```

```

        write(Reply),nl,
        check_reply(Id,Reply),!.

check_reply(Id,'?'):- explanation(Id),!,fail.
check_reply(Id,'y'):- action(Id).
check_reply(_, 'n').

/* The queries for input for the heat exchanger with no phase change case */

askable(fv,"Is the major disturbance a flow variation [y/n/?] ").
askable(cw," Is the utility stream cooling water [y/n/?] ").
askable(byp1," Is there an exchanger bypass on the controlled stream [y/n/?] ").
askable(byp2," Is there a bypass on the other stream [y/n/?] ").
askable(th1,"Can the controlled stream be throttled [y/n/?] ").
askable(th2,"Can the heating/cooling stream be throttled [y/n/?] ").
askable(th3,"Will the cooling water exit temperature be below 50 degrees C if the flow
is reduced below design [y/n/?] ").
askable(sv,"Is there sufficient driving force for a single bypass valve [y/n/?] ").
askable(tdp,"Is the operating temp > 260 degrees C or valve pressure drop high
[y/n/?]").

/* Queries for the steam heat exchanger case */

askable1(cl,"Is the exchanger designed to run with a condensate level [y/n/?] ").
askable1(hxv,"Is the exchanger mounted vertically [y/n/?] ").
askable1(byp3,"Is there a bypass on the temperature controlled stream [y/n/?] ").
askable1(hp,"Are there significant disturbances in steam header pressure [y/n/?] ").
askable1(ft1,"Are there significant disturbances in controlled stream temperature and
flow [y/n/?] ").
askable1(sv,"Is there sufficient driving force for a single bypass valve [y/n/?] ").
askable1(tdp,"Is the operating temp > 260 degrees C or valve pressure drop high
[y/n/?] ").
askable1(lcp,"Is the condensing pressure < condensate removal header pressure
[y/n/?]").
askable1(ovs,"Is the exchanger oversized for an expected operating condition
[y/n/?]").

/* Conditions for asking each query */

check_condition(fv):- dist(fast,_).
check_condition(cw):- duty(cool),htcool(ut).
check_condition(byp1).
check_condition(byp2):- not(target(byp)).
check_condition(th1):- not(bypass(av)).
check_condition(th2):- not(bypass(av)),not(target(tht)),not(htcool(cool)).
check_condition(th3):- not(bypass(av)),not(target(tht)),htcool(cool).
check_condition(sv):- bypass(av).
check_condition(tdp):- bypass(av).
check_condition(cl):- dist(slow,_).
check_condition(hxv):- dist(slow,large).
check_condition(byp3):- dist(fast,_).
check_condition(hp):- dist(fast,_),not(bypass(av)).
check_condition(ft1):- dist(fast,_),not(bypass(av)).
check_condition(lcp).
check_condition(ovs):- not(cr(lcp)).

/* Actions on a positive response to a query */

action(fv):- write("Select disturbed flow with arrow keys"),nl,
            menu(1,65,
            "Controlled",

```

```

    "Heating/cooling",
    "Cooling water"],
    Choice),insert1(Choice),write_flow(Choice).
action(cw):- asserta(htcool(cool)).
action(byp1):- asserta(target(byp)),asserta(bypass(av)).
action(byp2):- asserta(htcool(byp)),asserta(bypass(av)).
action(th1):- asserta(target(tht)).
action(th2):- asserta(htcool(tht)).
action(th3):- asserta(cool(tht)).
action(sv):- asserta(dp(av)).
action(tdp):- asserta(htdp(pr)).
action(cl):- asserta(shx(cond)).
action(hxv):- asserta(shx(v)).
action(byp3):- asserta(bypass(av)).
action(hp):- asserta(hp(v)).
action(ft1):- asserta(ff(shx)).
action(lcp):- asserta(cr(lcp)).
action(ovs):- asserta(cr(ovs)).

write_flow(Choice):- Choice=1,! ,write("The disturbed flow is the controlled
    stream"),nl.
write_flow(Choice):- Choice=2,! ,write("The disturbed flow is the heating/cooling
    stream"),nl.
write_flow(Choice):- Choice=3,! ,write("The disturbed flow is the cooling water
    stream"),nl.

/* Explanations for each of the queries */

explanation(fv):- write("If flow disturbances only are expected control can be
    improved by","\\n","cascading a primary temperature control loop onto flow
    control"),nl.
explanation(cw):- write(" Identifies the cooling method for later rules"),nl.
explanation(byp1):- write(" The preferred control method in this case is to bypass
    flow","\\n","around the exchanger."),nl.
explanation(byp2):- write("If there isn't a bypass on the controlled stream (better
    dynamics)","\\n"," then bypassing the heating/cooling stream flow is the
    accepted ","\\n"," control method in this case"),nl.
explanation(th1):- write(" Varying the controlled stream flowrate is a possible control
    method","\\n"," as long as downstream unit operations are unaffected by a
    changing feedrate"),nl.
explanation(th2):- write(" Varying the process heating/cooling stream flowrate is a
    possible","\\n","control method if downstream unit operations are unaffected
    by","\\n","changing feedrate"),nl.
explanation(th3):- write("If the cooling water flow is reduced to maintain temperature
    control","\\n","and its exit temperature exceeds 50 degrees C then accelerated
    ","\\n","heat exchanger fouling will occur."),nl.
explanation(sv):- write(" If a single valve is used in the bypass there must be
    sufficient","\\n","pressure drop to allow it to control"),nl.
explanation(tdp):- write(" If the temperature > 260 degrees C or there is a large
    pressure drop","\\n"," across the bypass valves a 3 way valve is unsuitable for
    the service"),nl.
explanation(cl):- write("If the exchanger is designed to run with a liquid level then the
    heat","\\n","transferred (and the exit temperature of the controlled stream) can
    be varied","\\n","by changing the liquid level"),nl.
explanation(hxv):- write("If there are large changes in heat load condensate level
    adjustment would","\\n","only successfully control temperature in a vertical
    exchanger"),nl.
explanation(byp3):- write("If a bypass is provided a good control option is to control
    temperature ","\\n","by bypassing some of the controlled stream flow (fast
    dynamics) and maintain ","\\n","the bypass valve on control by varying steam

```

```

condensing pressure via a valve", "\n", "position controller manipulating a steam
valve (gives the scheme good range)", nl.
explanation(hp):- write("This type of disturbance is handled effectively by cascading
temperature", "\n", "control onto steam condensing pressure or flow
control"), nl.
explanation(ft1):- write("Control can be improved by anticipating and correcting
feed", "\n", "flow/temperature disturbances using feed forward control"), nl.
explanation(lcp):- write("Specialised condensate removal techniques are required in
this case"), nl.
explanation(ovs):- write("Specialised condensate removal techniques are required in
this case"), nl.

```

```

insert(1):- asserta(dist(slow,small)).
insert(2):- asserta(dist(fast,small)).
insert(3):- asserta(dist(slow,large)).
insert(4):- asserta(dist(fast,large)).

```

```

insert1(1):- asserta(flow(tcon)).
insert1(2):- asserta(flow(htcool)).
insert1(3):- asserta(flow(cw)).

```

/* Rules for establishing the control type for heat exchangers without phase change */

```

contype:- dist(slow,_),target(tht),asserta(con(target,tht)),fail.
contype:- dist(slow,_),not(target(tht)),cool(tht),asserta(con(cw,tht)),fail.
contype:- dist(slow,_),not(target(tht)),not(htcool(cool)),htcool(tht),
    asserta(con(htcool,tht)),fail.
contype:- target(byp),asserta(con(target,byp)),fail.
contype:- htcool(byp),htcool(cool),asserta(con(cool,byp)),fail.
contype:- not(htcool(cool)),htcool(byp),asserta(con(htcool,byp)),fail.
contype:- dist(fast,_),flow(tcon),target(tht),asserta(con(target,tht_cascade)),fail.
contype:- dist(fast,_),flow(cw),not(target(tht)),cool(tht),
    asserta(con(cw,tht_cascade)),fail.
contype:- dist(fast,_),flow(htcool),not(target(tht)),htcool(tht),
    asserta(con(ht,tht_cascade)),fail.
contype:- dist(fast,_),target(tht),asserta(con(target,ff)),fail.
contype:- dist(fast,_),cool(tht),asserta(con(cw,ff)),fail.
contype:- dist(fast,_),htcool(tht),asserta(con(htcool,ff)),fail.
contype:- not(bypass(av)),not(con(_,_)),asserta(byp,t_one)),fail.
contype.

```

/* Rules for establishing control type for steam heat exchangers */

```

sh_contype:- dist(slow,small),shx(cond),not(cr(lcp)),asserta(con(cond,v_level)),fail.
sh_contype:- dist(slow,small),not(con(cond,v_level)),asserta(con(steam,tht)),fail.
sh_contype:- dist(fast,_),hp(v),not(ff(shx)),asserta(con(steam,tht_cascade)),fail.
sh_contype:- dist(fast,_),bypass(av),asserta(con(byp,vpc_st_tht)),fail.
sh_contype:- dist(fast,_),ff(shx),asserta(con(steam,ff)),fail.
sh_contype:- dist(slow,large),shx(cond),shx(v),not(cr(lcp)),
    asserta(con(cond,v_level)),fail.
sh_contype:- dist(slow,large),not(con(cond,v_level)),asserta(con(steam,tht)),fail.
sh_contype.

```

/* Rules for establishing condensate removal method */

```

cr_type:- dist(_,small),not(cr(lcp)),not(cr(ovs)),asserta(trap(n)),fail.
cr_type:- cr(lcp),asserta(trap(p)),fail.
cr_type:- dist(_,large),not(cr(ovs)),not(cr(lcp)),asserta(trap(d)),asserta(trap(lc)),fail.
cr_type:- cr(ovs),not(cr(lcp)),asserta(trap(d)),asserta(trap(lc)),fail.
cr_type.

```

```
/* Rules for establishing bypass type */
```

```
bypas:- bypass(av),dp(av),asserta(by(t_way)),fail.
bypas:- bypass(av),not(dp(av)),not(htdp(pr)),asserta(by(th_way)),fail.
bypas:- bypass(av),not(dp(av)),htdp(pr),asserta(by(t_t_way)),fail.
bypas.

byp_type:- bypass(av),write(" Recommended bypass type;"),nl,fail.
byp_type:- bypass(av),not(by(_)),write("I am unable to recommend a valve type from
      this information"),nl,!.
byp_type:- by(A),output_type1(A),fail.
byp_type.

con_meth:- write(" Recommended control types are ;"),nl,fail.
con_meth:- not(con(_,_)),write(" I am unable to recommend a control method from
      this information"),nl,!.
con_meth:- con(A,B),output_type(A,B),fail.
con_meth.

cond_removal:- write("Recommended condensate removal methods;"),nl,fail.
cond_removal:- not(trap(_)),write("I am unable to recommend a trap type from this
      information"),nl,!.
cond_removal:- trap(A),output_type2(A),fail.
cond_removal.
```

```
/* Control methods for heat exchangers without a phase change */
```

```
output_type(target,tht):- write(" Throttle the temperature controlled stream "),nl,!.
output_type(cw,tht):- write(" Throttle the cooling water stream"),nl,!.
output_type(htcool,tht):- write("Throttle the heating/cooling stream"),nl,!.
output_type(target,byp):- write("Bypass the temperature controlled stream"),nl,!.
output_type(cool,byp):- write("Bypass the cooling water stream"),nl,!.
output_type(htcool,byp):- write("Bypass the heating/cooling stream"),nl,!.
output_type(target,tht_cascade):- write("Cascade temp onto flow control on the
      temperature controlled stream"),nl,!.
output_type(cw,tht_cascade):- write("Cascade temp onto flow control of the cooling
      water stream"),nl,!.
output_type(ht,tht_cascade):- write("Cascade temperature onto flow control of the
      heating/cooling stream"),nl,!.
output_type(target,ff):- write("Use feed-forward control throttling the controlled
      stream"),nl,!.
output_type(cw,ff):- write("Use feed-forward control throttling the cooling water
      stream"),nl,!.
output_type(htcool,ff):- write("Use feed-forward control throttling the heating/cooling
      stream"),nl,!.
output_type(byp,t_one):- write(" Add a bypass around the exchanger for
      control"),nl,!.
```

```
/* Control types for steam heat exchangers */
```

```
output_type(cond,v_level):- write("Vary heat exchanger condensation area with a
      condensate valve"),nl,!.
output_type(steam,tht):- write("Vary heat exchanger condensing pressure with a steam
      valve"),nl,!.
output_type(steam,tht_cascade):- write("Vary heat exchanger condensing pressure
      with a steam valve. ","\n","Compensate for steam pressure upsets using
      temperature cascaded onto a","\\n","steam pressure or flow control loop"),nl,!.
output_type(byp,vpc_st_tht):- write("Control temperature using a bypass valve and
      keep this valve on","\\n","control using a valve position controller adjusting
      steam valve position"),nl,!.
```

```

output_type(steam,ff):- write("Vary heat exchanger condensing pressure with a steam
                           valve.", "\n", "Compensate for controlled stream temperature/flow variations
                           using", "\n", "feed forward control"), nl,!.

/* Bypass types */

output_type1(t_way):- write(" A single two way valve should be used in the
                           bypass"), nl,!.
output_type1(th_way):- write(" A three way valve should be used in the bypass"), nl,!.
output_type1(t_t_way):- write(" Two two-way valves should be used in the
                           bypass"), nl,!.

/* Condensate removal methods */

output_type2(n):- write("A normal thermodynamic trap will operate effectively"), nl,!.
output_type2(p):- write("A pumping trap is required to evacuate from the low
                           pressure"), nl,!.
output_type2(d):- write("A drainer trap is an option allowing level adjustment in the
                           exchanger"), nl,!.
output_type2(lc):- write("For very large load changes a level controlled vessel may be
                           required", "\n", "This provides a dual control method allowing both area change
                           and condensing", "\n", "pressure variation via the steam valve"), nl,!.

repeat.
repeat:- repeat.

cleanup:- retract(_), fail.
cleanup.

```


Appendix AVIII - TurboProlog Listing For The Reactor Expert System

1.0 Module REACT1

```
project "REACT"
include "globdef.pro"
domains
vr=symbol
str=string
ch=char
database - react1dom
dbase(vr,integer)ct(vr)
cons(vr)
ih(vr)
cat(vr)
lag(vr)
cm(vr)
fd(vr)
cv(vr)
cont(vr)
inertf(vr)
predicates
type
inform
askable(vr,str)
ask_user(vr,str)
check_condition(vr)
check_reply(vr,ch)
explanation(vr)
action(vr)
react
heat_ev
cool_type
cool_fluid
output_type(vr)
cont_type
set_up_window
run
unstable
fluid
goal
run.
clauses

run:- set_up_window,type,!,react,heat_ev,cool_type,cool_fluid,
      inform,cont_type,inv_comp_cont,cleanup.
run:- tub_fixed_bed.

type:- write("Is the reactor;\n",
```

AVIII-2

```

    "1) A CSTR\n",
    "2) A tubular or fixed bed reactor\n",
    "Enter the type [1/2] "),readint(X),X=1.

set_up_window:- makewindow(1,71,7,"Reactor Control Systems",0,0,25,80),
clearwindow.

react:- write("Is the reaction","\n",
    "1) exothermic ", "\n", "2) endothermic\n",
    "3) exothermic, feed preheat exceeds heat released by reaction\n",
    "Enter the choice [1/2/3] "),readint(X),nl,asserta(dbase(re,X)),fail.
react.

heat_ev:- dbase(re,1),write("Is the heat evolution/volume ", "\n", "1) small", "\n",
    "2) medium\n",
    "3) large", "\n", "Enter the choice [1/2/3] "),readint(X),
    asserta(dbase(he,X)),fail.
heat_ev.

cool_type:- dbase(re,1),not(dbase(he,3)),write("Does the CSTR have", "\n", "1) A
    jacket", "\n",
    "2) A coil\n",
    "3) No attached heat exchange", "\n", "As a means of cooling.",
    " Select the option that applies [1/2/3] "),readint(X),asserta(dbase(ct,X)),fail.
cool_type:- dbase(he,3),asserta(dbase(ct,3)),fail.
cool_type.

cool_fluid:- dbase(ct,1),write("Is the cooling fluid in the jacket", "\n",
    "1) Cooling water", "\n", "2) Heat transfer fluid", "\n", "3) Steam generation\n",
    "4) Cooling water or steam for initiating reaction\n",
    "Select the option that applies [1/2/3/4] "),readint(X),asserta(dbase(cf,X)),fail.
cool_fluid.

inform:- askable(Id,Question),ask_user(Id,Question),fail.
inform.
ask_user(Id,Question):- check_condition(Id),
    repeat,write(Question),
    readchar(Reply),
    write(Reply),nl,
    check_reply(Id,Reply),!.

check_reply(Id,'?'):- explanation(Id),!,fail.
check_reply(Id,'y'):- action(Id).
check_reply(_, 'n').

/* The queries for temperature control of Continuous Stirred Tank Reactors */

askable(ct,"Are variations in coolant supply temp expected [y/n/?] ").
askable(sl,"Are significant lags introduced by temp measurement, heat removal or
    reaction mass [y/n/?] ").
askable(ih,"Is heating required to initiate the reaction [y/n/?] ").
askable(cf,"Is catalyst feed available as a manipulated variable [y/n/?] ").
askable(cm,"Is the reaction mixture non-corrosive and not a slurry or polymer
    [y/n/?] ").
askable(fd,"Are fast dynamics needed in response to temperature fluctuations
    [y/n/?] ").
askable(cv,"Is reaction vapour condensed and recycled [y/n/?] ").
askable(vp,"Is there a continuous vapour product from a partial condenser [y/n/?] ").
askable(in,"Is there an inerts feed to the partial condenser [y/n/?] ").

/* Conditions that must be satisfied for the queries to be asked */

```

```

check_condition(sl):- dbase(he,2).
check_condition(ct):- dbase(re,1),not(dbase(cf,3)).
check_condition(ih):- unstable,dbase(ct,1),fluid.
check_condition(cf):- dbase(ct,3).
check_condition(cm):- dbase(ct,3).
check_condition(fd):- unstable,dbase(cf,2),dbase(ct,1),not(ih(y)).
check_condition(cv):- dbase(he,3).
check_condition(vp):- cv(pr).
check_condition(in):- cons(vp).

```

```

/* Actions taken on a positive response to the questions */

```

```

action(ct):- asserta(ct(vp)).
action(ih):- asserta(ih(y)).
action(cf):- asserta(cat(av)).
action(sl):- asserta(lag(pr)).
action(cm):- asserta(cm(n)).
action(fd):- asserta(fd(r)).
action(cv):- asserta(cv(pr)).
action(vp):- asserta(cons(vp)).
action(in):- asserta(inertf(pr)).

```

```

/* Explanations of the reasons for a question */

```

```

explanation(ct):- write("If variations in coolant temperature are expected control can be
    improved\n", "by cascading reaction temperature control onto a coolant
    temperature control loop"),nl.
explanation(ih):- write("Most exothermic reactions require heating to an ignition
    temperature before\n", "reaction occurs"),nl.
explanation(cf):- write("Variation of catalyst feed rate is a possibility for controlling
    temperature"),nl.
explanation(sl):- write("If the equipment has a medium heat evolution/vol and
    significant lags as\n", "described then open loop instability is likely"),nl.
explanation(fd):- write("When cooling fluid exiting the jacket is cooled by an external
    exchanger fast\n", "response is achieved using a bypass for control"),nl.
explanation(cm):- write("In this case the reaction mixture can be circulated through an
    external\n", "heat exchanger for cooling"),nl.
explanation(cv):- write("If there is vapour continuously boiling off the reaction mixture
    the vapour\n", "condensation rate can be altered to regulate reaction pressure
    and temperature"),nl.
explanation(vp):- write("If there is a continuous vapour product and inerts bleed to the
    condenser\n", "then split range control of these flows can regulate reaction
    pressure and temperature"),nl.
explanation(in):- write("If there is a continuous vapour product and inerts bleed to the
    condenser\n", "then split range control of these flows can regulate reaction
    pressure and temperature"),nl.

```

```

repeat.
repeat:- repeat.

```

```

/* Rules for establishing temperature control methods */

```

```

stable:- dbase(he,2),not(lag(pr)),!.
stable:- dbase(he,1).

unstable:- dbase(he,2),lag(pr),!.

temp_cont:- cont(h).

```

AVIII-4

```

fluid:- dbase(cf,1),!.
fluid:- dbase(cf,2).

cont_type:- dbase(ct,2),stable,not(ct(v)),asserta(cont(a)),fail.
cont_type:- dbase(ct,2),stable,ct(v),asserta(cont(act)),fail.
cont_type:- dbase(ct,1),stable,fluid,not(ct(v)),asserta(cont(b)),fail.
cont_type:- dbase(ct,1),stable,fluid,ct(v),asserta(cont(bct)),fail.
cont_type:- dbase(re,1),stable,asserta(cont(ft)),fail.
cont_type:- dbase(ct,1),dbase(cf,4),stable,asserta(cont(c)),fail.
cont_type:- dbase(ct,3),cat(av),asserta(cont(h)),fail.
cont_type:- dbase(ct,1),unstable,dbase(cf,1),not(ih(y)),asserta(cont(d)),fail.
cont_type:- dbase(ct,1),unstable,fluid,ih(y),asserta(cont(d1)),fail.
cont_type:- dbase(ct,3),cm(n),not(ct(v)),asserta(cont(f)),fail.
cont_type:- dbase(ct,3),cm(n),ct(v),asserta(cont(fct)),fail.
cont_type:- dbase(ct,3),not(dbase(he,3)),not(cm(n)),not(cat(av)),asserta(cont(ji)),fail.
cont_type:- dbase(ct,1),dbase(cf,2),unstable,not(ih(y)),fd(r),asserta(cont(e2)),fail.
cont_type:- dbase(ct,1),dbase(cf,2),unstable,not(ih(y)),not(fd(r)),
    asserta(cont(e1)),fail.
cont_type:- dbase(ct,1),dbase(cf,3),unstable,asserta(cont(i)),fail.
cont_type:- dbase(he,3),cv(pr),not(cons(vp)),not(inertf(pr)),asserta(cont(g2)),fail.
cont_type:- dbase(he,3),cv(pr),cons(vp),inertf(pr),asserta(cont(g1)),fail.
cont_type:- dbase(ct,2),unstable,asserta(cont(cu)),fail.
cont_type:- dbase(re,2),asserta(cont(en)),fail.
cont_type:- dbase(re,3),asserta(cont(psex)),fail.
cont_type.

/* Output different control types */

cont_meth:- write("Recommended control types are; "),nl,fail.
cont_meth:- not(cont(_)),write("I am unable to recommend a control method from this
    information "),nl,!.
cont_meth:- cont(A),output_type(A),fail.
cont_meth.

/* Description of temperature control methods */

output_type(a):- write("Control temperature by throttling the coolant flow in the
    cooling coil"),nl,!.
output_type(act):- write("Control temperature by throttling the coolant flow in the
    cooling coil\n","Cascade reaction temperature onto a coolant temperature
    control loop"),nl,!.
output_type(b):- write("Control temperature by throttling coolant flow to the
    jacket"),nl,!.
output_type(ft):- write("Control reaction temperature by varying feed
    temperature"),nl,!.
output_type(bct):- write("Control temperature by throttling coolant flow to the
    jacket\n","Cascade reaction temperature onto a coolant temperature control
    loop"),nl,!.
output_type(c):- write("Control temperature by throttling coolant flow to the
    jacket\n","Initial heat-up provided by steam to the jacket"),nl,!.
output_type(d):- write("Use a pump-around of coolant through the jacket. Control
    temperature\n","by adjusting coolant make-up rate. Hot coolant is discharged
    from the circuit\n","under pressure control"),nl,!.
output_type(d1):- write("Use a pump-around of coolant through the jacket. Control
    temperature\n","by adjusting coolant make-up rate. Hot coolant is discharged
    from the circuit\n","under pressure control. Use a steam heated exchanger in
    the circuit for\n","initial heating"),nl,!.
output_type(e1):- write("Use a pump-around of coolant through the jacket with an
    external heat\n","exchanger. Control coolant jacket temperature by throttling the
    flow on the\n","cooling side of the heat exchanger. Cascade reaction
    temperature control onto\n","this loop"),nl,!.

```

```

output_type(e2):- write("Use a pump-around of coolant through the jacket with an
    external heat\n","exchanger. Control coolant jacket inlet temperature by
    bypassing coolant flow\n","around the exchanger. Cascade reaction
    temperature control onto this loop"),nl,!.
output_type(f):- write("Control temperature by circulating the reaction mixture through
    an\n","external heat exchanger. Throttle the flow on the cooling side of this\n",
    "exchanger to control reaction mixture temperature"),nl,!.
output_type(fct):- write("Control temperature by circulating the reaction mixture
    through an\n","external heat exchanger.Cascade the reaction temperature
    control loop onto\n","a coolant temperature control loop"),nl,!.
output_type(g2):- write("Control temperature by varying condenser cooling rate"),nl,!.
output_type(g1):- write("Control temperature by a split range controller on vapour
    product\n","flow and inerts bleed to the partial condenser"),nl,!.
output_type(h):- write("Control temperature by varying catalyst feed rate"),nl,!.
output_type(i):- write("Control temperature by varying steam generation pressure in
    the jacket\n","Cascade onto a pressure control loop. Feedwater supplied to the
    jacket on\n","level control"),nl,!.
output_type(en):- write("Since this is an endothermic reaction temperature control
    isn't\n","critical. Place heat source on flow control and monitor reaction
    temperature"),nl,!.
output_type(ji):- write("Try a CSTR wuth a jacket to obtain a controllable
    reactor"),nl,!.
output_type(cu):- write("A CSTR with a coil has insufficient heat removal capacity to
    stabilise\n","this reaction. Try a CSTR with a jacket"),nl,!.
output_type(psex):- write("Normally temperature control would be by varying the
    heating\n","rate but if the heat release exceeds feed preheat in abnormal
    operation\n","cooling is required. The safest scheme is a split range
    temperature\n","controller manipulating heating rate or cooling rate as
    required"),nl,!.

```

2.0 Module CSTR

```

include "globdef.pro"
domains
vr=symbol
str=string
ch=char
database - cstrdom
dbasec(vr,integer)
prod(vr)
con(vr)
conc(vr)
obj(vr)
cf(vr)
st(vr)
ic(vr)
fr(vr)
react(vr)
b(vr)
stor_b(vr)
stor_s(vr)
clean(vr)
predicates
inform
askable(vr,str)
ask_user(vr,str)
check_condition(vr)
check_reply(vr,ch)
explanation(vr)

```

AVIII-6

```

action(vr)
conv_meth
conc_meth
con_type
reactor_type
feed_type
prod_type
output_type(vr)
output_type1(vr)
clauses
inv_comp_cont:- reactor_type,feed_type,prod_type,
                inform,con_type,cont_meth,conv_meth,conc_meth,cleanup.

/* This section inputs the relevant details for the rules */

reactor_type:- write("Is the reactor;\n",
                    "1) A single-pass reactor\n",
                    "2) Part of a recycle loop\n",
                    "Enter the type [1/2] "),readint(X),asserta(dbasec(typ,X)),fail.
reactor_type.

feed_type:- dbasec(typ,1),write("Is there feeding the reactor;\n",
                    "1) A single liquid feed\n",
                    "2) Two liquid feeds\n",
                    "3) A gas feed and a liquid feed\n",
                    "Select the option that applies [1/2/3] "),readint(X),asserta(dbasec(fe,X)),fail.

feed_type:- dbasec(typ,2),write("Is the feed to the recycle reactor;\n",
                    "1) Two pure component liquid feeds\n",
                    "2) Two pure liquid feeds, reaction moderated by a recycled solvent\n",
                    "Select the type [1/2] "),readint(X),asserta(dbasec(recy,X)),fail.
feed_type:- dbasec(fe,2),write("Is one or both feeds available for control [1/2] "),
                readint(X),asserta(dbasec(fc,X)),fail.
feed_type.

prod_type:- dbasec(fe,3),write("Is the product stream a gas or liquid [gas/liq] "),
                readln(Var), asserta(prod(Var)),fail.
prod_type.

inform:- askable(Id,Question),ask_user(Id,Question),fail.
inform.

ask_user(Id,Question):- check_condition(Id),
                        repeat,write(Question),
                        readchar(Reply),
                        write(Reply),nl,check_reply(Id,Reply),!.

check_reply(Id,'?'):- explanation(Id),!,fail.
check_reply(Id,'y'):- action(Id).
check_reply(_, 'n').

/* These are the conditions for information requests */

check_condition(cn):- dbasec(typ,1).
check_condition(co):- dbasec(typ,1),not(obj(con)).
check_condition(ic):- dbasec(typ,1).
check_condition(fr):- dbasec(typ,1),dbasec(fe,1),obj(com),ic(y).
check_condition(cf):- dbasec(typ,1),obj(com),ic(y).
check_condition(ex):- dbasec(typ,2),dbasec(recy,1).
check_condition(sb):- dbasec(typ,2),dbasec(recy,1),b(ex).
check_condition(cs):- dbasec(typ,2),dbasec(recy,1),b(ex).

```

```
check_condition(ss):- dbasec(typ,2),dbasec(recy,2).
check_condition(so):- dbasec(typ,2),dbasec(recy,2).
```

```
askable(cn,"Is conversion a control objective [y/n/?] ").
askable(co,"Is exit composition a control objective [y/n/?] ").
askable(ic,"Are variations expected in reactant feed composition [y/n/?] ").
askable(fr,"Is the feedrate available for control [y/n/?] ").
askable(cf,"Is catalyst feed available as a manipulation [y/n/?] ").
askable(ex,"Is one of the reactants (B) in excess [y/n/?] ").
askable(sb,"Is there storage available for recycled reactant B [y/n/?] ").
askable(cs,"Does the separation in the recycle loop return pure B [y/n/?] ").
askable(ss,"Is there storage available for the recycled solvent [y/n/?] ").
askable(so,"Is reactant recycled with the solvent after separation from the products
[y/n/?] ").
```

```
/* These are the alterations to the database taken on a positive response to queries */
```

```
action(cn):- asserta(obj(con)).
action(co):- asserta(obj(com)).
action(ic):- asserta(ic(y)).
action(fr):- asserta(fr(av)).
action(cf):- asserta(cf(av)).
action(ex):- asserta(b(ex)).
action(sb):- asserta(stor_b(av)).
action(cs):- asserta(clean(b)).
action(ss):- asserta(stor_s(av)).
action(so):- asserta(react(so)).
```

```
/* Explanations of the queries from the program */
```

```
explanation(cn):- write("Control methods for constant conversion vary from those
required\n", "for constant composition"),nl.
explanation(ic):- write("If inlet composition varies then exit composition will also
vary\n", "even when conversion remains constant"),nl.
explanation(co):- write("Different control methods are required to keep composition
rather than\n", "composition constant"),nl.
explanation(fr):- write("Feedrate variations can be used to control composition
when reactor\n", "holdup is controlled"),nl.
explanation(cf):- write("If there is a catalyst feed to the reactor it can be used for
composition\n", "control"),nl.
explanation(ex):- write("If one of the reactants is in excess of stoichiometry it must
be\n", "separated from the products and recycled "),nl.
explanation(sb):- write("The proposed control system requires storage for the excess
reactant\n", "for recycle"),nl.
explanation(cs):- write("The proposed control system requires that only excess reactant
and not\n", "product be recycled"),nl.
explanation(ss):- write("Recycle solvent storage is necessary"),nl.
explanation(so):- write("If reactants are soluble in the recycled solvent control
adjustments\n", "must be made to maintain reactor exit composition"),nl
```

```
/* Rules for product inventory control */
```

```
con_type:- dbasec(typ,1),not(dbasec(fe,3)),asserta(con(lc)),fail.
con_type:- dbasec(typ,1),dbasec(fe,3),prod(gas),asserta(con(gp)),fail.
con_type:- dbasec(typ,1),dbasec(fe,3),prod(liq),asserta(con(lp)),fail.
con_type:- dbasec(typ,2),dbasec(recy,1),b(ex),stor_b(av),clean(b),
asserta(con(recy1)),fail.
con_type:- dbasec(typ,2),dbasec(recy,2),stor_s(av),asserta(con(recy2)),fail.
```

```
/* Rules for composition and condition control */
```

```

con_type:- dbasec(typ,1),dbasec(fe,1),obj(con),asserta(conc(fc1)),fail.
con_type:- dbasec(typ,1),dbasec(fe,1),obj(com),not(ic(y)),asserta(conc(fc1)),fail.
con_type:- dbasec(typ,1),dbasec(fe,1),obj(com),ic(y),fr(av),asserta(conc(fc3)),fail.
con_type:- dbasec(typ,1),dbasec(fe,2),dbasec(fc,2),obj(con),asserta(conc(fc2)),fail.
con_type:- dbasec(typ,1),dbasec(fe,2),dbasec(fc,2),obj(com),not(ic(y)),
    asserta(conc(fc2)),fail.
con_type:- dbasec(typ,1),dbasec(fe,2),dbasec(fc,2),obj(com),ic(y),
    asserta(conc(fct_t)),fail.
con_type:- dbasec(fc,1),obj(_),asserta(conc(sf)),fail.
con_type:- dbasec(typ,1),dbasec(fe,3),obj(con),asserta(conc(tp)),fail.
con_type:- dbasec(typ,1),dbasec(fe,3),obj(com),not(ic(y)),asserta(conc(tp)),fail.
con_type:- dbasec(typ,1),obj(com),ic(y),stable,asserta(conc(tc)),fail.
con_type:- dbasec(typ,1),obj(com),ic(y),cf(av),not(temp_cont),asserta(conc(cf)),fail.
con_type:- dbasec(typ,2),con(recy1),asserta(conc(recy1)),fail.
con_type:- dbasec(typ,2),con(recy2),not(react(so)),asserta(conc(recy1)),fail.
con_type:- dbasec(typ,2),con(recy2),react(so),asserta(conc(recy2)),fail.
con_type.

```

/* This section outputs the inventory control types */

```

conv_meth:- write("Recommended inventory control types are;"),nl,fail.
conv_meth:- not(con(_)),write("I am unable to recommend a control scheme from this
    information"),nl,fail.
conv_meth:- con(A),output_type(A),fail.
conv_meth.

```

/* This section outputs the composition control types */

```

conc_meth:- write("Recommended composition control types are;"),nl,fail.
conc_meth:- not(conc(_)),write("I am unable to recommend a control scheme from this
    information"),nl,fail.
conc_meth:- conc(A),output_type1(A),fail.
conc_meth.

```

/* Descriptions of the different inventory control methods */

```

output_type(lc):- write("The product should be on level control"),nl,!.
output_type(gp):- write("The liquid feed should be added on level control. The gas
    feed sparged\n", "in on flow control and the product withdrawn on pressure
    control. There\n", "should be a manual liquid bleed to remove any accumulated
    inerts"),nl,!.
output_type(lp):- write("The liquid feed should be on flow control. The gas feed
    sparged in on\n", "pressure control and the product withdrawn on level control.
    There should be\n", "a manual gas purge to remove accumulated inerts"),nl,!.
output_type(recy1):- write("Feed the limiting and recycle reactants on flow control.
    Make up\n", "the consumed excess reactant on level control to the recycle
    storage\n", "tank. The product should be on level control"),nl,!.
output_type(recy2):- write("Feed the reactants on flow control (with ratio control
    between the\n", "flows). Recycle the solvent from storage on flow control. The
    product\n", "should be on level control"),nl,!.

```

/* Descriptions of the different composition control types */

```

output_type1(fc1):- write("Place feed on flow control keeping residence time
    constant"),nl,!.
output_type1(fc2):- write("Both feeds on flow control with ratio control between
    them"),nl,!.
output_type(fc2_t):- write("Both feeds on flow control with ratio control between
    them\n", "Trim the ratio using an exit composition controller"),nl,!.
output_type(sf):- write("Place the available feed on ratio control to the other feed. \n",
    "Calculate changes in setpoint for a level controller on the product\n",

```



```

        "to maintain spacetime constant. Override this signal with a low \n"
        and a remote controller if the level exceeds a limit value "),nl,!.
output_type1(fc3):- write("Vary feedrate to control residence time "),nl,!.
output_type1(tc):- write("Control composition by varying reation temperature and
        reaction rate"),nl,!.
output_type1(cf):- write("Control composition by varying catalyst feedrate"),nl,!.
output_type1(tp):- write("The proposed inventory control scheme will keep residence
        time and\n","conversion constant"),nl,!.
output_type1(recy1):- write("The inventory control scheme will maintain composition
        constant in\n","this case"),nl,!.
output_type1(recy2):- write("Because of the recycled reactant the ratio controller
        between the\n","reactant feed flows should be trimmed by feedback from a
        composition\n","controller to maintain exit composition"),nl,!.

/* This predicate cleans out the database */

cleanup:- retract(_),fail.
cleanup.

```

3.0 Module TUBFBR

```

project "REACT"
include "globdef.pro"
domains
vr=symbol
database - tubfbrdom
dbaset(vr,integer)
q(vr)
cat(vr)
dt(vr)
pb(vr)
in(vr)
pf(vr)
con(vr)
conr(vr)
conv(vr)
predicates
react
reactor_type
feed_type
r_type
cool_fluid
check_type
inform
askable(vr,string)
ask_user(vr,string)
check_reply(vr,char)
check_condition(vr)
explanation(vr)
action(vr)
heat_ev
stable1
con_type
conr_type
conv_type
con_meth
conv_meth
conr_meth
output_type(vr)

```

clauses

```
/* Input section for the program */
```

```
tub_fixed_bed:- react,heat_ev,reactor_type,feed_type,r_type,cool_fluid,inform,
con_type,conr_type,conv_type,con_meth,conv_meth,conr_meth,cleanup.
```

```
react:- write("Is the reaction;\n",
           "1) Exothermic\n","2) Endothermic\n",
           "Enter the type [1/2] "),readint(X),nl,
           asserta(dbaset(re,X)),fail.
react.
```

```
heat_ev:- dbaset(re,1),write("Is the heat evolution/volume ", "\n","1) small", "\n",
           "2) medium\n",
           "3) large", "\n","Enter the choice [1/2/3] "),readint(X),
           asserta(dbaset(he,X)),fail.
heat_ev.
```

```
reactor_type:- write(" Is the reactor;\n",
                    "1) A single pass reactor\n",
                    "2) Part of a recycle loop\n",
                    "Enter the type [1/2] "),readint(X),asserta(dbaset(typ,X)),fail.
reactor_type.
```

```
feed_type:- write("Is there feeding the reactor;\n",
                  "1) A single feed\n",
                  "2) Two feeds\n",
                  "Select the feedtype [1/2] "),readint(X),nl,
                  asserta(dbaset(fe,X)),fail.
feed_type.
```

```
r_type:- write("Is the reactor;\n",
               "1) A jacketed pipe\n",
               "2) A heat exchanger type reactor (with or without catalyst in the tubes)\n",
               "3) A multi-bed fixed bed reactor\n",
               "4) A fired heater\n",
               "Enter the type [1/2/3/4] "),readint(X),nl,
               asserta(dbaset(r_t,X)),fail.
r_type.
```

```
cool_fluid:- dbaset(re,1),check_type,write("Is the cooling fluid in the jacket\n",
      "1) Coolant flow\n",
      "2) Steam raised in the jacket\n",
      "Enter the type [1/2] "),readint(X),asserta(dbaset(ct,X)),fail.
cool_fluid.
```

```
check_type:- dbaset(r_t,1),!.
check_type:- dbaset(r_t,2).
```

```
inform:- askable(Id,Question),ask_user(Id,Question),fail.
inform.
```

```
ask_user(Id,Question):- check_condition(Id),
                        repeat,write(Question),
                        readchar(Reply),
                        write(Reply),nl,
                        check_reply(Id,Reply),!.
```

```
check_reply(Id,"?"):- explanation(Id),!,fail.
```

```

check_reply(Id,'y'):- action(Id).
check_reply(_, 'n').

askable(sl,"Are significant lags introduced by temp measurement, heat removal or
reaction mass [y/n/?] ").
askable(ft,"Is the feed temperature before preheating << reaction temperature
[y/n/?] ").
askable(cf,"Is catalyst feed available as a manipulated variable [y/n/?] ").
askable(dt,"Is the feed flowrate or composition variable [y/n/?] ").
askable(pb,"Is a purge bleed available as a manipulated variable [y/n/?] ").
askable(in,"Do inerts enter the the recycle loop with the feed [y/n/?] ").
askable(inv,"Is the feed inerts concentration variable [y/n/?] ").
askable(pf,"Is the purge flowrate measured [y/n/?] ").
askable(vc,"Are variations expected in the flow or temperature of the coolant [y/n/?] ").

check_condition(sl):- dbaset(he,2).
check_condition(ft):- dbaset(re,1),dbaset(r_t,3).
check_condition(cf):- dbaset(re,1),dbaset(ct,_).
check_condition(dt):- cat(av).
check_condition(pb):- dbaset(typ,2).
check_condition(in):- pb(av).
check_condition(inv):- in(f).
check_condition(pf):- in(v).
check_condition(vc):- check_type,dbaset(ct,1).

explanation(sl):- write("If the reaction has a medium and significant lags as\n"
"described then open loop instability is likely"),nl.
explanation(ft):- write("A quench temperature significantly lower than reaction
temperature\n", "is required for this control method"),nl.
explanation(cf):- write("Catalyst feed rate can be used to control reaction rate if
available"),nl.
explanation(dt):- write("Reaction rate control is only necessary when there are feed
variations"),nl.
explanation(pb):- write("In a recycle loop or mixed phase product reactor purge
flowrate can be\n", "used to control loop (or reactor) temperature"),nl.
explanation(in):- write("Inerts must be purged from the loop"),nl.
explanation(inv):- write("If the amount of inerts entering the loop varies then purge rate
also\n", "changes (if it is on pressure control)"),nl.
explanation(pf):- write("The inerts concentration in the loop can be kept constant by
adjusting\n", "feed rate to hold purge rate constant"),nl.

action(sl):- asserta(lg(pr)).
action(ft):- asserta(q(av)).
action(cf):- asserta(cat(av)).
action(dt):- asserta(dt(y)).
action(pb):- asserta(pb(av)).
action(in):- asserta(in(f)).
action(inv):- asserta(in(v)).
action(pf):- asserta(pf(m)).
action(vc):- asserta(vc(y)).

stable1:- dbaset(he,2),not(lg(pr)),!.
stable1:- dbaset(he,1).

/* Rules for the control of temperature in tubular reactors */

con_type:- dbaset(re,1),check_type,stable1,dbaset(ct,1),asserta(con(a)),fail.
con_type:- con(a),vc(y),asserta(con(vc)),fail.
con_type:- dbaset(re,1),dbaset(r_t,1),dbaset(ct,2),not(dbaset(he,3)),
asserta(con(b)),fail.

```

```

con_type:- dbaset(re,1),dbaset(r_t,2),dbaset(ct,2),asserta(con(b)),fail.
con_type:- dbaset(re,2),dbaset(r_t,4),asserta(con(e)),fail.

/* Rules for the control of temperature in fixed bed reactors */

con_type:- dbaset(re,1),dbaset(r_t,3),q(av),asserta(con(c)),fail.
con_type:- dbaset(re,1),dbaset(r_t,3),not(q(av)),asserta(con(d)),fail.
con_type:- dbaset(re,2),dbaset(r_t,3),asserta(con(f)),fail.

/* Rules for exceptional circumstances */

con_type:- dbaset(re,1),dbaset(he,3),check_type,dbaset(ct,1),asserta(con(t_hx)),fail.
con_type:- dbaset(re,1),dbaset(he,3),dbaset(r_t,1),dbaset(ct,2),asserta(con(t_hx)),fail.
con_type:- dbaset(re,1),dbaset(he,2),lg(pr),check_type,dbaset(ct,1),
    asserta(con(t_hx)),fail.
con_type:- dbaset(re,2),check_type,asserta(con(na)),fail.
con_type.

/* Rule for reaction rate control */

conr_type:- con(b),cat(av),dt(y),asserta(conr(g)),fail.
conr_type.

/* Inventory control rules */

conv_type:- dbaset(fe,1),asserta(conv(a1)),fail.
conv_type:- dbaset(fe,2),asserta(conv(b1)),fail.
conv_type:- dbaset(typ,2),pb(av),asserta(conv(a2)),fail.
conv_type:- dbaset(typ,2),pb(av),in(f),in(v),pf(m),asserta(conv(b2)),fail.
conv_type.

/* This section outputs the temperature control types */

con_meth:- write("Recommended temperature control types are;"),nl,fail.
con_meth:- con(A),output_type(A),fail.
con_meth.

/* This section outputs the inventory control types */

conv_meth:- write("Recommended inventory control methods are;"),nl,fail.
conv_meth:- conv(A),output_type(A),fail.
conv_meth.

/* This section outputs the rate control method */

conr_meth:- conr(_),write("Recommended rate control method;"),nl,fail.
conr_meth:- conr(A),output_type(A),fail.
conr_meth.

/* Descriptions of the control types */

output_type(a):- write("Control temperature by throttling coolant flow to the
    jacket"),nl,!.
output_type(vc):- write("Cascade the primary temperature controller onto a secondary
    jacket\n","temperature controller"),nl,!.
output_type(b):- write("Control temperature by adjusting the steam pressure in the
    jacket"),nl,!.
output_type(c):- write("Control bed inlet temperatures by varying quench flowrates to
    the beds"),nl,!.
output_type(d):- write("Control bed inlet temperatures by varying interbed heat
    exchanger cooling rate"),nl,!.

```

```

output_type(e):- write("Control outlet temperature from the furnace by varying firing
rate"),nl,!.
output_type(f):- write("Control bed inlet temperature by varying interbed heat
exchanger heating rate"),nl,!.
output_type(g):- write("Control reaction rate using steam rate as the control objective
and\n","manipulate catalyst feed to the reactor"),nl,!.
output_type(a1):- write("Place feed to the reactor on flow control "),nl,!.
output_type(a2):- write("Control pressure in the recycle loop by manipulating purge
bleed rate"),nl,!.
output_type(b1):- write("Feeds to the reactor should be on flow control with ratio
control\n","between the two flows"),nl,!.
output_type(b2):- write("Control loop pressure by manipulating purge flowrate. Vary
the feedrate\n","in response to changes in purge flowrate caused by fluctuations
in inerts\n","concentration in the loop"),nl,!.
output_type(t_hx):- write(" The heat release of this reaction requires a heat
exchanger\n","type reactor with steam raised in the shell for cooling"),nl,!.
output_type(na):- write("The program has no rules to handle endothermic reactions
in\n","this type of reactor. Try a fixed bed reactor or fired heater"),nl,!.

```


Appendix AIX - TurboProlog Listings of The Extra Programs For The Coordinated Expert Systems

1.0 Coordination Routine for the Individual Expert Systems

```
project "SYNCON"
include "globdef1.pro"
database - processbase
elim(integer,vrlist)
calc_order(integer,vr)
man(integer,vrlist)
man_used(integer,vrlist)
elimted(vrlist)
predicates
run(integer)
choose_opn(integer)
cont_system(integer)
check_elim(integer)
mod(integer,vrlist)
find_cont(integer,vr)
p_chek(vrlist)
p_enter(vrlist,vrlist,vrlist)
check(vr,vr,vrlist)
v_u_else(integer,vrlist,vrlist,vrlist)
contl(vrlist)
r_p(vrlist)
del_list(vrlist,vrlist,vrlist)
elim_fail(vrlist)
ch_member(vrlist,vrlist,integer,integer)
mod_m(integer,vrlist,vrlist)
all_paired(vrlist)
set_up_window
goal
consult("test2"),
consult("test1",processbase),
set_up_window,
run(1).
clauses

set_up_window:- makewindow(1,71,7,"Control System Synthesis",0,0,25,80),
clearwindow.

/* Controls program execution */

run(X):-choose_opn(X),repeat,cont_system(X),check_elim(X),X1=X+1,
man_used(X,A),mod(X1,A),!,run(X1).
run(_):- pair(X,Y),write("Pair ",X," with ",Y),nl,fail.
```

```
/* Chooses unit operation by calculation order */
```

```
choose_opn(X):- calc_order(X,Type),find_cont(X,Type).
```

```
/* Finds control system alternatives for a unit operation */
```

```
find_cont(X,dist):- distillation(X),!.
find_cont(X,hx):- heat_exchange(X),!.
find_cont(X,react):- reactor(X),!.
find_cont(X,_):- unknown(X).
```

```
/* Allows entry of the selected unit operation control system */
```

```
cont_system(X):- name(X,B),write("Enter the pairing information for the
    unpaired\n","control objectives in unit operation ",B),
    nl,cont_obj(X,A),p_chek(A),all_paired(A),man(X,C),
    p_enter(A,TheList,C),asserta(man_used(X,TheList)),!.
```

```
/* Checks whether a pairing has already been made for a control objective */
```

```
p_chek([]).
p_chek([H|T]):- pair(H,Y),!,write(H," must be paired with ",Y),nl,p_chek(T).
p_chek([_|T]):- p_chek(T).
```

```
/* Ensures that the entered MVs can be used in that unit operation */
```

```
check(H,Y,B):- path(H,Y),member(Y,B),!.
check(H,Y,_):- write("The manipulated variable ",Y," either isn't available in this\n",
    "unit operation or doesn't affect control objective ",H,". Try again "),nl,fail.
```

```
/* Enters the selected pairing information */
```

```
p_enter([],[],_).
p_enter([H|T],[Y|T1],B):- not(pair(H,_)),!,repeat,
    write("Enter the manipulated variable paired with control objective ",H," "),
    readln(Y),check(H,Y,B),asserta(pair(H,Y)),p_enter(T,T1,B),!.
p_enter([H|T],[Y|T1],B):- pair(H,Y),!,p_enter(T,T1,B).
```

```
/* Checks whether all the control objectives are paired in a unit operation */
```

```
all_paired([]):- !,write("All the control objectives are paired in this unit operation"),nl.
all_paired([H|T]):- pair(H,_),!,all_paired(T).
all_paired(_).
```

```
/* Checks whether eliminated variables fail controllability data */
```

```
check_elim(X):-
    man_used(X,M),man(X,B),del_list(M,B,L1),v_u_else(X,L1,L1,L2),
    contl(L2),!.
check_elim(X):- retract(man_used(X,M)),!,r_p(M),fail.
```

```
/* Removes pairs for a failed control system */
```

```
r_p([]).
r_p([H|T]):- retract(pair(_,H)),!,r_p(T).
```

```
/* Checks if variables can be used in a unit operation not calculated yet */
```

```
v_u_else(_,[],L3,L3).
```



```

v_u_else(X,[H|T],L1,L3):- man(Y,M),Y>X,member(H,M),!,del(H,L1,L),
    v_u_else(X,T,L,L3).
v_u_else(X,[_|T],L1,L3):- v_u_else(X,T,L1,L3).

/* Checks a units eliminated variables against the plant elimination information */

contl([]):- !.
contl(L):- elimted(N),append(L,N,L1),not(elim_fail(L1)),
    retract(elimted(N)),!,asserta(elimted(L1)).

/* Succeeds if the variables violate the plant elimination information */

elim_fail(L1):- elim(Z,E),ch_member(L1,E,0,Z).

/* Counts up the number of variables in a test list that belong to an elimination list */

ch_member([],_,I,Z):- !,Z<I.
ch_member([H|T],E,I,Z):- member(H,E),!,NewI=I+1,ch_member(T,E,NewI,Z).
ch_member([_|T],E,I,Z):- ch_member(T,E,I,Z).

/* Removes MVs used from units yet to be calculated */

mod(X,A):- man(X,B),!,mod_m(X,A,B),X1=X+1,mod(X1,A).
mod(_,_).

mod_m(X,[],B):- retract(man(X,_)),asserta(man(X,B)),!.
mod_m(X,[H|T],B):- member(H,B),!,del(H,B,B1),mod_m(X,T,B1).
mod_m(X,[_|T],B):- mod_m(X,T,B).

/* These are prolog utility routines required by the program */

member(R,[R|_]).
member(R,[_|Tail]):- member(R,Tail).

append([],B,B).
append([X|L1],List2,[X|L3]):- append(L1,List2,L3).

del_list([],B,B).
del_list([H|T],B,B1):- del(H,B,L),del_list(T,L,B1).

del(X,[X|T],T):-!.
del(X,[Y|T],[Y|T1]):- del(X,T,T1).

repeat.
repeat:- repeat.

```

2.0 Program for Handling Unknown Unit Operations

```

project "SYNCON"
include "GLOBDEF1.PRO"
predicates
remove_cont(vrlist,vrlist,vrlist)
controllable(vrlist,vrlist)
reverse(vrlist,vrlist)
reverse1(vrlist,vrlist,vrlist)
writelist1(vrlist)
writelist2(vrlist)
wl(vrlist)
wll(vr)
clauses

```

```

unknown(X):- name(X,C),write("Control possibilities for ",C), nl,
              cont_obj(X,B),remove_cont(B,B,B1), reverse(B1,B2),
              writelist1(B2),nl,controllable([],B1),fail.
unknown(_).

remove_cont([],B,B).
remove_cont([H|T],B,B1):- pair(H,_),!,del(H,B,L),remove_cont(T,L,B1).
remove_cont([_|T],B,B1):- remove_cont(T,B,B1).

controllable(B,[]):- writelist2(B).
controllable(B,[H|A]):- path(H,Y),not(pair(_,Y)),not(member(Y,B)),
                        append([Y],B,L),controllable(L,A).

writelist1([]):- !,fail.
writelist1(List):- write("Control Objectives "),wl(List),nl.

writelist2(List):- write("
                        "),wl(List),nl.

wl([]):- !.
wl([First|Rest]):- !,wll(First),wl(Rest).
wll(C):- write(C," ").

reverse(L,R):- reverse1(L,[],R).
reverse1([],R,R).
reverse1([H|R],W,R1):- reverse1(R,[H|W],R1).

```